

ClaimCenter New and Changed Guide

Release 6.0.8



Copyright © 2001-2013 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Live, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This material is Guidewire proprietary and confidential. The contents of this material, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

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

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Product Name: Guidewire ClaimCenter

Product Release: 6.0.8

Document Name: ClaimCenter New and Changed Guide

Document Revision: 05-February-2013

Contents

About This Document	21
Intended Audience	21
Assumed Knowledge	21
Related Documents	21
Conventions In This Document	22
Support	22

Part I

What's New and Changed in 6.0

1 New and Changed in ClaimCenter 6.0	25
New in ClaimCenter 6.0	25
Catastrophe Bulk Association	25
Claims Performance Monitoring	26
Deductible Handling	26
Electronic Fund Transfers (EFT)	26
ICD Codes	26
Integrating with PolicyCenter	26
New Travel Line of Business	26
Policy Currency	26
Reinsurance	27
Workers' Compensation Contains New Features	27
New System Permissions	28
Changes in ClaimCenter 6.0	29
Administration Tab Contains New Items	29
Advanced Searches	29
Business Weeks	30
Currency Aware Authority Limit Profile	30
FNOL Wizards Are Updated	30
Homeowner's Screens Have Been Expanded	30
Integration with Policy Administration Systems	30
ISO Expanded	30
Workers' Compensation Screens Have Been Reorganized	30
Minor Exposure and Typelist Method Changes	31

2	New and Changed in Configuration in 6.0	33
	Changes to the Guidewire ClaimCenter Data Model	33
	New Configuration Parameter Access	34
	Added, Removed, or Changed Configuration Parameters	34
	Localized Database Columns	35
	New Language Column on Document and Activity Entities	36
	Reference Activity Pattern by Code, Not ID	36
	New Metadata File Structure	36
	Changed FieldValidators File	37
	Changes to Data Types	38
	Changes to Entity Types	39
	Changes to Data Objects	39
	Changes to Database Tables	40
	Removal of <validationTriggerOverrides> Element	40
	Contact Casting Accessors Deprecated	40
	Column Attributes Become columnParam Subelements	40
	Columns Added to HolidayZone Entity	41
	Removed Entity Properties	41
	Renamed Entity Properties	42
	Changes in Studio	42
	Changes to Studio Dialogs	42
	Changes to Studio Editors	43
	Changes to Studio Menus	43
	Changes to Studio Right-Click Contextual Menus	43
	Changes Related to PCF Files	44
	PCF Schema Changes	44
	Server Start Up No Longer Verifies PCF Files	44
	Changes Related to Workflow	44
	Improvements in Localization	45
	Set a Locale for Studio	45
	Localize Specific Activity Pattern Fields	45
	Localize Field Validation	45
	Set Zone Information	45
	Localize Documents, Notes, and Templates	46
	Localize Document APIs	46
	Localized Field Validation	47
	Japanese Imperial Calendar Date Picker	47
	Localization of Command Line Tools Argument Descriptions	47
	Localization of Workflow	47
	Catastrophe Bulk Associations Batch Job	48
	Changes in Duplicate Claim and Check Searches	48
	Configuring Claim Health Metrics	48
	Configuring Deductibles	48
	Configuring Approval Rules with Bulk Invoice Checks	48
	Configuring Recently Viewed Claims in the Claim Tab	49

3	New and Changed in Gosu in 6.0	51
	New in Gosu in 6.0	51
	Gosu Shell	52
	Gosu Generated Documentation from Type System	52
	Run Local Command-line Commands from Gosu	52
	Shell-related APIs	52
	Increment and Decrement Operators	52
	New Expression Operators	53
	New Compound Assignment Operators	53
	Gosu Templates	54
	Gosu Composition Syntax	55
	Object Lifecycle Management with the 'using' Keyword	56
	Type Inference Downcasting	56
	Profiler Tags	57
	The Object Equality Operator (==)	57
	Object Initializer Syntax During Object Creation	58
	Type System Reflection New APIs	59
	New Stream Utilities	59
	Concurrency Utilities	59
	Checksum APIs	59
	Run With New Bundle With a User	59
	Changed in Gosu in 6.0	60
	Renamed GScript to Gosu	60
	Moved Classes Hierarchy	60
	Java Collection Generics Fixes	60
	Data Structure Declaration Extensions for Lists, Maps, and Sets	61
	Class Variables Can Both Expose as Property and Have Initializer	61
	Compound Types	62
	Change in Ternary Conditionals If Clauses Return Different Types	62
	Moved Enhancement Packages	63
	XML Node Package Name Changed	63
	Gosu Error Reporting Improvements	63
	Improvements to XSD and XML Processing	63
	Changes to XSD Class Loading Behavior	64
	Query Builder API Changes	65
	Query Builder API Improvements	65
	Blocks Require Single Statements in Braces	66
	Changes to Existing Collections Enhancement Methods	67
	New Collections Enhancement Methods	68
	New Array/List Expansion Operator (Deprecated Old Style)	71
	Gosu Array Enhancement Changes	72
	New Concurrency and Scoping APIs, Scoped Variables Deprecated	73
	New 'Type' Property on All Types	73
	Exception Changes If No Current User and Creating New Bundle	74
	Block Declarations Now Require Argument Names	74
	Function Pointers and Nested Functions Now Unsupported	74
	Annotation Syntax Change for Run Time Access	75
4	New and Changed in System Administration in 6.0	77
	New in System Administration in 6.0	77
	Batch Process to Purge Completed Workflows	77
	Graph Validation Checks	77

Changed in System Administration in 6.0	78
Application Server Requirements	78
Database Server Requirements	78
Operating System Requirements	78
Java and ANT Versions	78
Caching Mechanism	78
Administrative Data Export Granularity	79
5 New and Changed in Integration in 6.0	81
New in Integration in 6.0	81
Tools for XML Export of Types for Integration	81
Asynchronous Document Store and Transport	82
Startable Plugins	82
Create Custom Batch Processes	82
Profiler Web Services	82
Simple Servlets	82
Field-level Encryption	83
Claim Search (PolicyCenter Integration)	83
Authentication User Role Syncing	83
Large Loss Notification and Policy System Notification Architecture	84
New Preupdate Handler Plugin	84
Backup Withholding Plugin	84
Changes in Integration in 6.0	85
Changes to Structure of Integration-related Files and Scripts	85
Copy the Plugin Entity Library to Your Configuration Module	86
ISO Changes in ClaimCenter 6.0	87
FNOL Mapper is Now a Server-Side Tool	94
Financials Integration APIs for Acknowledgement-based Transitions	95
Validation Plugin Removed	96
Multicurrency-related Changes	97
Claim Financials Web Service API Changes	98
Messaging API Changes	98
Document Management Plugin Implementation Location Changes	101
Parameterization of Types Stripped from Java External Entities	101
New Plugin Registry (Old Version Deprecated in Gosu)	101
IMessagingToolsAPI Web Service Interface Changes	102
IDataExtractionAPI Web Service Interface Deprecated	102
IClaimAPI Web Service Interface Changes	103
IClaimFinancialsAPI Web Service Interface Changes	103
Document Management Plugin Changes for Availability	103
Geocoding Plugin Changes for Map Options	104
Testing Clock (For Development Use Only)	104
Summary of Plugin Changes, Additions, and Removals	104
Web Service Changes, Additions, and Removals	105
Authentication Integration Changes	106
Enforcement of Web Service Type Name Conflicts	106
Metropolitan Timeout Changes	106
Segmentation Plugin Deprecated	107
6 New and Changed in Rules in 6.0	109
New in Rules in ClaimCenter 6.0	109
Create Activity Pattern by Code Rather Than by ID	109

Changes in Rules in ClaimCenter 6.0	110
ClaimCenter Stores Rules as Gosu Classes	110
Rule Set Category Name Changes	111
Rule Set Category Creation through Studio	111
Modified setApprovingUser Method	111
Removed Rules	111
Rule Debugger Menu Changes	112
Resource Export and Import Removed	112
7 New and Changed in Reporting in 6.0.	113
New in Reporting in ClaimCenter 6.0	113
Changes in Reporting in ClaimCenter 6.0	114

Part II

What's New and Changed in 5.0

8 New and Changed in ClaimCenter 5.0.	119
New in ClaimCenter 5.0	119
Address Completion	120
Assessments	120
Catastrophes	121
Claim Archiving	121
Claim Associations	122
Injury Incidents	122
Lines of Business	122
Multiple Currencies.	122
Question Sets.	123
QuickJump Box.	123
Regional Holidays.	123
Security on Notes	123
Service Provider Management	123
Subrogation	124
User Interface	124
Workflows.	125
Zone Mapping and Autofill.	125

Changes in ClaimCenter 5.0	125
Significant Assignment Changes	126
Calendar Enhancements	128
Configuration Parameters	128
Coverage Model Changes	128
Coverage Verification	128
Data and Security Dictionaries	128
Database Consistency Checks	128
Eclipse No Longer Supported	129
Email Enhancements	129
Export	129
Financial Changes	129
Geocoding	129
Localization	129
Matter and Negotiation Enhancements	130
Module-Based Configuration	130
New Claim Wizard	130
New Work Queue Parameters	130
Parties Involved Enhancements	131
PCF Definition	131
PCF Input Widgets	131
Policy Model Changes	131
SIU Questionnaire	131
Workflow Flexibility	131
9 New and Changed in GScript in 5.0	133
New in GScript in 5.0	133
GScript Annotations	133
GScript Blocks	133
GScript Generics	134
Publishing GScript Code as New SOAP APIs	134
Calling SOAP APIs Published by External Applications	134
Native GScript Support for XML and XSDs	134
GScript Enhancements	134
GScript List Enhancements: Sort, Find, Map, Each	134
JUnit Testing	134
New GScript Try/Catch Syntax	135
GScript Interface Support	135
GScript Inner Classes	135
GScript Abstract, Override, and Final Modifiers	135
GScript Transaction APIs	136
GScript Enumerations	136
Collections and String Size and Length Improvements	136
List and Map Indexing Improvements	136
List and Map In-line Initialization Improvements	137
Improved Java Interopability	137

Changed in GScript in 5.0.	137
GScript Constructor Syntax Changed.	137
Stricter Return Statements Requirements for All Code Paths	138
Java Type Information Changes	139
EntityFactory Deprecated in GScript (Not in Java)	139
GScript Libraries Changes	139
GScript Case Sensitivity Implications	140
Java Properties Capitalization Changes	140
Deprecation of itype Property	141
Minor Change In Meaning of 'List'	141
Typekey.Name Becomes DisplayName and UnlocalizedName.	142
Type.TypeKeys Property Becomes Method with Argument	142
TypeList.getByCode() Changes	143
Type Information Exposed for All Types	143
Removal of toMap Method on Collection	143
Change in Equality Comparison, and New Warnings for Edge Cases	144
10 New and Changed in Installation in 5.0.	147
New in Installation in 5.0	147
QuickStart Installation	147
Support for WebLogic JNDI Data Source	147
Changed in Installation in 5.0.	148
SQL Server Parameter Required.	148
Updated JVM Heap Size Considerations	148
Integration with ContactCenter Simplified.	148
11 New and Changed in Integration in 5.0.	149
New in Integration in 5.0	149
Major New SOAP Features in GScript.	149
New Email APIs, Including Document Management.	149
EntityFactory in Java Method Name Change	149
Multicurrency Integration Support	149
SOAP API Non-Conversational Mode Now Supported.	150
Rendering Arbitrary Input Stream Data, Such as PDF Data.	150
Database Staging Table Import Conversion View	150
Messaging Threads Configurable	150

Changes in Integration in 5.0	151
IDataObjectAPI is Deprecated	151
Plugins Directories Moved In Relation to Config Directory	153
Web Service Visibility and Customizability Changes	154
Regenerating the Toolkit Changes	154
Plugin Registration Changes	155
Plugin Built-in Implementation Classes Package Changes	155
Messaging Destination Registration Moved	156
GWServices Plugins Removed	156
GScript Plugin Important and Required Changes	156
SOAP Plugins No Longer Needed	158
IAssignmentAdapter Deprecated	159
IExportTools Removed	159
Geocoding Plugin Changes	160
ISO Configuration Changes	162
Document Management Changes	163
Message State and Message History Changes	164
Messaging Transaction Changes and MessageRequest Plugin	165
Ensure Messaging Code Does Not Rely on Assignment Entity	165
Message Sinks Now Unsupported	166
Transaction Public ID Length Must Be Shorter	166
Multi-Registration of Plugins for Wrapping Unsupported	166
Contact Autosync	167
Changes to Parameter Order in Outgoing SOAP API Requests	167
MapPoint Updates for New WSDL Parameter Ordering	168
Changes to Message Ordering and Multi-Threading	168
12 New and Changed in Rules in 5.0	173
New in Rules in 5.0	173
Archive Rules	173
Changes in Rules in 5.0	173
Assignment Rules	174
Exposure Get Recoveries API Returns Read-only Entities	175
13 New and Changed in Studio in 5.0	177
New in Studio in 5.0	177
Standalone (Disconnected) Operation	177
Studio Resource Management	178
Studio Interface Changes	178
New Configuration Options	178
Graphical Studio Editors	180
GUnit Tester	182
New GScript API Documentation Support	182
Keyboard Shortcuts	182
New Contextual Right-Click Menu Commands	182
Delete (or Revert to Base) Files	182
View the SCM Log	183
View and Edit (More) Files in Studio	183
New Text Editor	183
New Help Menu PCF Format Reference	183

Changes in Studio in 5.0	183
New Development Mode	184
Resource View Is Virtual	184
Manage Most Resources from Studio	184
Library Functions Become Classes and Enhancements	185
Studio Options Moved to Tools Menu	185
Studio Handles Script Parameters Differently	185
Studio Links to SCM System	186

Part III

Release Notes Archive

14 Guidewire ClaimCenter 4.0.0 Release Notes	189
Overview	189
Release Information	189
Version Number	189
Installation Requirements	190
Support	190
Installing This Release	190
Changes in this Release	190
Known Issues and Limitations	190
15 Guidewire ClaimCenter 4.0.1 Release Notes	193
Overview	193
Release Information	193
Version Number	193
Installation Requirements	194
Support	194
Installing This Release	194
Changes in this Release	194
Configuration Environment Changes	194
PCF Syntax Changes	195
PCF Implementation Notes	197
Improvements	200
General Issues	200
Known Issues and Limitations	201
16 Guidewire ClaimCenter 4.0.2 Release Notes	205
Release Information	205
Version Number	205
Installation Requirements	205
Support	205
Installing This Release	206
Notes on Upgrading Rules from Release 4.0.x	206
Changes in this Release	206
Configuration Environment Changes	206
PCF Syntax Changes	208
Base PCF File Changes	209
Base Rule Changes	209
Improvements	209
General Issues	211
Known Issues and Limitations	212
17 Guidewire ClaimCenter 4.0.3 Release Notes	217
Overview	217

Release Information	218
Version Number	218
Installation Requirements	218
Support	218
Installing This Release	218
ClaimContact converts into ISO ClaimParty elements.	218
Changes in this Release	219
Configuration Environment Changes	220
PCF Syntax Changes.	222
Base PCF File Changes.	222
Base Rule Changes	222
Improvements	223
General Issues	225
Known Issues and Limitations	229
18 Guidewire ClaimCenter 4.0.4 Release Notes	233
Overview.	233
Release Information	234
Version Number	234
Support	234
Installing This Release	234
An AssignmentStatus Column Change May Require Upgrading.	234
A New Plugin Method is Required.	234
You Must Manually Upgrade the Rule Sets	234
Acknowledgement Changes	235
Changes in this Release	235
Configuration Environment Changes	235
PCF Syntax Changes.	237
Base PCF File Changes.	237
Base Resource Changes	237
Improvements	238
General Issues	239
Known Issues and Limitations	242
19 Guidewire ClaimCenter 4.0.5 Release Notes	247
Overview.	247
Release Information	248
Version Number	248
Support	248
Installing This Release	248
Changed Event Filtering Behavior For Messaging.	248
Manually Upgrade the Rule Sets (CC-39604)	249
How to Use itype and typeof (CC-39157)	249
Staging Table Import (CC-35141)	249
Install a New Reporting Module.	250
Best Practices for Upgrading Rules and Libraries	250
Changes in this Release	250
Configuration Environment Changes	251
Base PCF File Changes.	251
Base Resource Changes	251
Improvements	252
General Issues	252
Known Issues and Limitations	254

20	Guidewire ClaimCenter 4.0.6 Release Notes	261
	Overview	261
	Release Information	261
	Version Number	262
	Support	262
	Installing This Release	262
	Changes to ISO Validation Rules	262
	Manually Upgrade the Rule Sets (CC-39604)	262
	Best Practices for Upgrading Rules and Libraries	262
	Changes in this Release	263
	Configuration and GScript API Changes	263
	Base PCF File Changes	264
	Base Resource Changes	264
	Improvements	265
	General Issues	265
	Known Issues and Limitations	266
21	Guidewire ClaimCenter 4.0.7 Release Notes	271
	Overview	271
	Release Information	271
	Version Number	272
	Support	272
	Installing This Release	272
	Manually Upgrade the Rule Sets (CC-39604)	272
	Best Practices for Upgrading Rules and Libraries	272
	Changes in this Release	273
	Configuration and GScript API Changes	273
	Base PCF File Changes	273
	Base Resource Changes	273
	Improvements and General Issues	274
	Known Issues and Limitations	275
22	Guidewire ClaimCenter 4.0.8 Release Notes	281
	Overview	281
	Release Information	281
	Version Number	282
	Support	282
	Installing This Release	282
	Manually Upgrade the Rule Sets (CC-39604)	282
	Best Practices for Upgrading Rules and Libraries	282
	Changes in this Release	283
	Configuration and GScript API Changes	283
	Base PCF File Changes	283
	Base Resource Changes	283
	Improvements and General Issues	284
	Known Issues and Limitations	286
23	Guidewire ClaimCenter 4.0.9 Release Notes	293
	Overview	293
	Release Information	293
	Version Numbers	294
	Support	294
	InetSoft Evaluation	294

Installing This Release	294
Software Updates	294
Manually Upgrade the Rule Sets (CC-39604)	294
Best Practices for Upgrading Rules and Libraries	295
Changes in this Release	295
Configuration and GScript API Changes	295
Base PCF File Changes	296
Base Resource Changes	296
Improvements and General Issues	297
Known Issues and Limitations	298
24 Guidewire ClaimCenter 5.0.0 Release Notes	307
Overview	307
Release Information	307
Version Number	308
Support	308
Installing This Release	308
Changes in this Release	308
Known Issues and Limitations	308
25 Guidewire ClaimCenter 5.0.1 Release Notes	317
Overview	317
Release Information	317
Version Number	318
Support	318
Installing This Release	318
Changes in this Release	318
Configuration and GScript API Changes	318
Base PCF File Changes	318
Base Resource Changes	318
getFieldValue/setFieldValue Changes (CC-42508)	319
TypeInfo Subobjects Changed Package (CC-41379)	322
New TypeSystem Class (CC-41379)	322
Changes to Integration Entity Libraries in 5.0.1 (CC-41164)	322
Messaging Plugins and IBaseURLBuilder Plugin Changes (CC-42481)	324
Geocoding Plugin Changes (CC-39806)	325
Changing Coverage Triggers Policy Changed Event (CC-29564)	325
Advanced Authentication to ContactCenter	325
Improvements	326
General Issues	328
Known Issues and Limitations	329
Documentation-Related Issues	337
Manually merging WC- and MP-prefixed typelists (CC-42663)	337
ABAddress Renamed to Address	337
Upgrading Libraries and the Java Method invokeLibraryMethod	337
26 Guidewire ClaimCenter 5.0.2 Release Notes	339
Overview	339
Release Information	339
Version Number	340
Support	340
Installing This Release	340
Note Regarding Rules After Upgrade (CC-44608)	340

Changes in this Release	340
Configuration and GScript API Changes	340
Base PCF File Changes	340
Base Resource Changes	341
Improvements	341
General Issues	342
Known Issues and Limitations	347
27 Guidewire ClaimCenter 5.0.3 Release Notes	355
Overview	355
Release Information	355
Version Number	356
Support	356
Installing This Release	356
Changes in this Release	356
Configuration and GScript API Changes	356
Base PCF File Changes	356
Base Resource Changes	356
Configuration Parameter Changes	357
Improvements and General Issues	357
Known Issues and Limitations	361
28 Guidewire ClaimCenter 5.0.4 Release Notes	369
Overview	369
Release Information	369
Version Number	370
Support	370
Installing This Release	370
Inetsoft Evaluation	370
Wrapping of Labels	370
Method Removed	370
Changes to Parameter Order in Outgoing SOAP API Requests	370
MapPoint Updates	371
Changes in this Release	372
Configuration and GScript API Changes	372
Base PCF File Changes	372
Base Resource Changes	372
Configuration Parameter Changes	372
Improvements and General Issues	373
Known Issues and Limitations	377
29 Guidewire ClaimCenter 5.0.5 Release Notes	387
Overview	387
Release Information	387
Version Number	388
Support	388
Installing This Release	388
Inetsoft Evaluation	388
Software Updates	388
Archiving Updates	388
Changes to HTML Tables	389

Changes in this Release	389
Configuration and GScript API Changes	389
Base PCF File Changes	389
Base Resource Changes	389
Configuration Parameter Changes	389
Improvements and General Issues	390
Known Issues and Limitations	394
30 Guidewire ClaimCenter 5.0.6 Release Notes	407
Overview	407
Release Information	407
Version Number	408
Support	408
Installing This Release	408
Inetsoft Evaluation	408
Upgrade Issues	408
Software Updates	408
Archiving Updates	408
Changes to HTML Tables	409
Changes in this Release	409
Configuration and GScript API Changes	409
Base PCF File Changes	410
Base Resource Changes	410
Improvements and General Issues	410
Known Issues and Limitations	416
31 Guidewire ClaimCenter 5.0.7 Release Notes	429
Overview	429
Release Information	429
Version Number	430
Support	430
Installing This Release	430
Inetsoft Evaluation	430
Software Updates	430
Archiving Updates	430
Changes in this Release	431
Base PCF File Changes	431
Base Resource Changes	431
Improvements and General Issues	431
Known Issues and Limitations	434
32 Guidewire ClaimCenter 5.0.8 Release Notes	449
Overview	449
Release Information	449
Version Number	450
Support	450
Installing This Release	450
Inetsoft Evaluation	450
Software Updates	450
Archiving Updates	450
Changes in this Release	451
Base PCF File Changes	451
Base Resource Changes	451
Improvements and General Issues	451
Known Issues and Limitations	453

33	Guidewire ClaimCenter 6.0.0 Release Notes	467
	Overview	467
	Release Information	467
	Version Number	468
	Support	468
	Installing This Release	468
	Inetsoft Evaluation	468
	Upgrade Issues	468
	Reporting Issues	469
	Changes in this Release	469
	Configuration and GScript API Changes	469
	Base PCF File Changes	469
	Known Issues and Limitations	470
34	Guidewire ClaimCenter 6.0.1 Release Notes	479
	Overview	479
	Release Information	479
	Version Number	480
	Support	480
	Installing This Release	480
	InetSoft Evaluation	480
	Issues	480
	Changes in this Release	481
	Base PCF File Changes	481
	Rules Changes	481
	Improvements and General Issues	481
	Known Issues and Limitations	492
35	Guidewire ClaimCenter 6.0.2 Release Notes	503
	Overview	503
	Release Information	503
	Version Number	504
	Support	504
	Installing This Release	504
	InetSoft Evaluation	504
	Issues and Major Changes	504
	Changes in this Release	505
	Base PCF File Changes	506
	Rules Changes	506
	Improvements and General Issues	506
	Display Key Changes for Localized Modules	515
	Known Issues and Limitations	519
36	Guidewire ClaimCenter 6.0.3 Release Notes	531
	Overview	531
	Release Information	531
	Version Number	532
	Support	532
	Installing This Release	532
	InetSoft Evaluation	532
	Issues and Major Changes	532

Changes in this Release	533
Base PCF File Changes	533
Rules Changes	534
Improvements to Upgrade Diff Report	534
Improvements and General Issues	534
Known Issues and Limitations	538
37 Guidewire ClaimCenter 6.0.4 Release Notes	551
Overview	551
Release Information	551
Version Number	552
Support	552
Installing This Release	552
InetSoft Evaluation	552
Issues and Major Changes	552
Changes in this Release	555
Base PCF File Changes	555
Rules Changes	555
Improvements to Upgrade Diff Report	555
Improvements and General Issues	556
Known Issues and Limitations	559
ClaimCenter Known Issues	559
Platform/Studio Known Issues	564
38 Guidewire ClaimCenter 6.0.5 Release Notes	573
Overview	573
Release Information	574
Version Number	574
Installing This Release	574
InetSoft Evaluation	574
Issues and Major Changes	574
Support	575
Changes in this Release	575
Base PCF File Changes	576
Rules Changes	576
Improvements to Upgrade Diff Report	576
Improvements and General Issues	576
Known Issues and Limitations	583
ClaimCenter Known Issues	583
Platform/Studio Known Issues	588
39 Guidewire ClaimCenter 6.0.6 Release Notes	593
Overview	593
Release Information	594
Version Number	594
Installing This Release	594
InetSoft Evaluation	594
Support	594
Changes in this Release	594
Base PCF File Changes	595
Rules Changes	595
Improvements to Upgrade Diff Report	595
Improvements and General Issues	595

Known Issues and Limitations	599
ClaimCenter Known Issues	600
Platform/Studio Known Issues	605
40 Guidewire ClaimCenter 6.0.7 Release Notes	609
Overview	609
Release Information	610
Version Number	610
Installing This Release	610
InetSoft Evaluation	610
Support	610
Changes in this Release	610
Base PCF File Changes	611
Rules Changes	611
Changes in This Release Provided in Upgrade Diff Report	611
Updating Localization modules	611
Improvements and General Issues	611
Known Issues and Limitations	615
ClaimCenter Known Issues	616
Platform/Studio Known Issues	622

About This Document

This document describes new features and changes to existing features in ClaimCenter 6.0. This document focuses on significant changes between major releases of ClaimCenter. To read about changes between minor releases of ClaimCenter, refer to the release notes for that release.

This topic includes:

- “Intended Audience” on page 21
- “Assumed Knowledge” on page 21
- “Related Documents” on page 21
- “Conventions In This Document” on page 22
- “Support” on page 22

Intended Audience

This document is intended for the following readers:

- Business users who want to upgrade to ClaimCenter 6.0.8 and understand the new features and changes in this release.
- System administrators

Assumed Knowledge

This document assumes that you are already familiar with the following topics:

- Using previous releases of ClaimCenter.

Related Documents

See the following Guidewire documents for further information:

ClaimCenter Upgrade Guide – Provides instructions to upgrade ClaimCenter.

ClaimCenter Installation Guide – Describes how to install a new copy of ClaimCenter into Windows or UNIX environments. This guide is intended for system administrators and developers who need to install ClaimCenter.

ClaimCenter System Administration Guide – Provides guidance for the ongoing management of a ClaimCenter system. This document is intended to help system administrators monitor ClaimCenter, manage its security, and take care of routine tasks such as system backups, logging, and importing files.

ClaimCenter Application Guide – Introduces the application, explains application concepts, and provides a high-level view of major features and business goals of ClaimCenter. This is your first place to look when learning about a feature. This book is written for all audiences.

Gosu Reference Guide – Describes the syntax of expressions and statements within ClaimCenter. This document also provides examples of how the syntax is used when creating rules. This document is intended for rule writers who create and maintain rules in Guidewire Studio.

ClaimCenter Integration Guide – Provides an architectural overview and examples of how to integrate ClaimCenter with external systems and custom code. This document is a learning tool for explanations and examples with links to the *Java API Reference Javadoc* and *SOAP API Javadoc* for further details. This document is written for integration programmers and consultants.

ClaimCenter Data Dictionary – Describes the ClaimCenter data model, including your custom data model extensions. To generate the dictionary, go to the ClaimCenter/bin directory and run the gwcc regen-dictionary command. To view the dictionary, open the ClaimCenter/build/dictionary/data/index.html file. For more information about generating and using the *Data Dictionary*, see the *ClaimCenter Configuration Guide*.

Conventions In This Document

Text style	Meaning	Examples
<i>italic</i>	Emphasis, special terminology, or a book title.	A <i>destination</i> sends messages to an external system.
bold	Strong emphasis within standard text or table text.	You must define this property.
narrow bold	The name of a user interface element, such as a button name, a menu item name, or a tab name.	Next, click Submit .
monospaced	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.	Get the field from the Address object.
<i>monospaced italic</i>	Parameter names or other variable placeholder text within URLs or other code snippets.	Use getName(<i>first</i> , <i>last</i>). http://SERVERNAME/a.html.

Support

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

- At the Guidewire Resource Center – <http://guidewire.custhelp.com>
- By email – support@guidewire.com
- By phone – +1-650-356-4955

part I

What's New and Changed in 6.0

New and Changed in ClaimCenter 6.0

New in ClaimCenter 6.0

This topic describes new features in ClaimCenter 6.0. These include:

- “Catastrophe Bulk Association” on page 25
- “Claims Performance Monitoring” on page 26
- “Deductible Handling” on page 26
- “Electronic Fund Transfers (EFT)” on page 26
- “ICD Codes” on page 26
- “Integrating with PolicyCenter” on page 26
- “New Travel Line of Business” on page 26
- “Policy Currency” on page 26
- “Reinsurance” on page 27
- “Workers’ Compensation Contains New Features” on page 27
- “New System Permissions” on page 28

Catastrophe Bulk Association

Carriers closely track their total exposure for catastrophes because they often have reinsurance agreements that cover their exposure over a certain amount. With ClaimCenter 6.0, you can mark every claim that is due to a catastrophe, even if the claim was created *before* the catastrophe profile was set up in ClaimCenter. You do this in the **Administration** tab, by searching for claims that match the catastrophe profile, but have not yet been linked to it and linking the claim from there.

For detailed information, see “Catastrophe Bulk Association” on page 118 in the *Application Guide*.

Claims Performance Monitoring

ClaimCenter provides a set of tools that monitors the health of a claim and automatically tracks key metrics within each individual claim. These tools enable adjusters and managers to make a quick diagnosis as to whether a claim is being processed efficiently or whether it is in need of additional consideration.

- **Claim Health Metrics** are embedded within every claim and track key metrics and compares them to configurable company-specific benchmarks.
- **High Risk Indicators** are visible in an enhanced **Claim Summary** screen and persistent on the claim Info Bar. They provide a continual reminder of the special circumstances surrounding a claim.
- **The Claim Headline** at the top of the **Claim Summary** screen presents a view of the most important aspects of a claim's overall condition.
- **Claim Reports** aggregate key claim information into logical groupings. Managers can take appropriate action based on the information contained in the reports.

See "Claim Performance Monitoring" on page 315 in the *Application Guide*.

Deductible Handling

ClaimCenter supports deductible handling for the personal auto line of business.

See "Deductible Handling" on page 183 in the *Application Guide* for details.

Electronic Fund Transfers (EFT)

You can perform financial transactions electronically. These are known as electronic funds transfer or EFT. In ClaimCenter, you work with EFTs mainly in the context of a contact.

See "Electronic Funds Transfer (EFT)" on page 148 in the *Application Guide*.

ICD Codes

The *International Statistical Classification of Diseases and Related Health Problems* (ICD) are medical diagnosis codes that are accepted standards worldwide. You edit the codes in the **Administration** tab, and apply them through the Medical Diagnosis field in lines of businesses where there is an injury.

See "ICD Codes" on page 417 in the *Application Guide* for details.

Integrating with PolicyCenter

The default configuration of ClaimCenter includes a functional integration with Guidewire PolicyCenter. (You must integrate this.) You can also integrate ClaimCenter with the policy system of your choice. Integrations include viewing policies in a policy administration system and large loss notifications.

See "Policy System Integration" on page 335 in the *Application Guide*.

New Travel Line of Business

ClaimCenter 6.0 offers a new line of business.

See "Personal Travel Line of Business" on page 289 in the *Application Guide*.

Policy Currency

ClaimCenter 6.0 has broadened the use of multicurrency to support carriers who write policies in more than one country. Specifically, this feature aids carriers who have policies that span multiple countries and have different

currencies. You can manage the claims for all of those countries in one instance of ClaimCenter. The new currency is *policy currency* and is one of several currencies that ClaimCenter supports.

The currency types are:

- *Default*, which is sometimes called reporting currency. This is set in the configuration parameter `DefaultApplicationCurrency` in the `config.xml` file.
- *Policy*, which gets its currency from the `Policy` entity. ClaimCenter copies the `Currency` field in `Policy` to the `Claim` entity. This means that the policy and the claim currency are always the same.
- *Transactional*, which is the currency of the transaction amount. This is the primary amount for the transaction, from which other amounts are calculated. For payments, this is the currency in which the actual payment was made.

For a complete description, see “Multiple Currencies” on page 171 in the *Application Guide*.

Reinsurance

Reinsurance is the process of insurance companies insuring underwritten policies with other institutions in order to offset exposure. Carriers do this to offset some of the financial risks that insurance companies assume when they insure any kind of loss type. This relationship is called a reinsurance *agreement* or *treaty*, but is not a *policy*.

See “Reinsurance” on page 269 in the *Application Guide*.

Workers’ Compensation Contains New Features

The workers’ compensation line of business contains the following new features:

Compensability Decision

Compensability decision refers to discovering and deciding if a workers’ compensation claim is valid, and hence payable. In ClaimCenter, you can decide this based on jurisdiction. See “Compensability Decision” on page 303 in the *Application Guide*.

Jurisdictional Benefit Calculation Management

In ClaimCenter, you can calculate the payments for lost time. The key calculations involve:

- **TPD - Temporary Partial Disability**: the jurisdictional maximum to pay the injured worked each week; may or may not be dependent on the `BaseRate`.
- **TTD - Temporary Total Disability**: the percentage of the `BaseRate` that is paid to injured workers as their benefit.
- **PPD - Permanent Partial Disability**: the jurisdictional minimum to pay the injured worked each week; may or may not be dependent on the `BaseRate`.
- **PTD - Permanent Total Disability**: A common exception to lower the Jurisdictional Minimum Comp Rate. If the `BaseRate` is lower than the mandated minimum, then states with this exception lower the jurisdictional minimum to the `BaseRate`.

See “Jurisdictional Benefit Calculation Management” on page 305 in the *Application Guide* for details.

Finding Injured Workers

Since a company can have more than one injured worker, ClaimCenter can be configured to allow adjusters to sort by injured worker.

See “Finding Injured Workers” on page 304 in the *Application Guide*.

See Also

- “Workers’ Compensation Line of Business” on page 297 in the *Application Guide*.

New System Permissions

ClaimCenter 6.0 includes new system permissions. These permissions are presented by code along with a short description. For more information about a particular permission, see the *ClaimCenter Security Dictionary*. See “Regenerating the Data Dictionary and Security Dictionary” on page 32 for instructions to generate the *ClaimCenter Security Dictionary*.

Code	Permission to...
buswkmanage	create, edit, or delete business weeks
buswkview	view the list of business weeks
delmednote	delete a medical note
editdeductible	edit deductibles
editmednote	edit a medical note
editrefdata	edit administration reference data
metriclimitmanage	create, edit, and delete claim and exposure metric limits
orgcreate	create organizations
orgdelete	delete organizations
orgeditbasic	edit basic information about organizations
orgsearch	search for organizations
orgviewbasic	view basic information about organizations
reinstthresholdedit	edit reinsurance reportable thresholds
reinstthresholdview	view reinsurance reportable thresholds
reinsuranceedit	edit reinsurance reportable status on a claim
toolsBatchProcessedit	edit the Batch Process Info Server Tools page
toolsBatchProcessview	access the Batch Process Info Server Tools page
toolsCacheinfoview	view the Cache Info Server Tools page
toolsClusteredit	edit the Cluster Internal Tools page
toolsClusterview	access the Cluster Internal Tools page
toolsInfoview	access the Info Pages Internal Tools page
toolsJMXBeansEdit	edit the Management Beans Server Tools page
toolsJMXBeansview	access the Management Beans Server Tools page
toolsJProfileredit	edit the JProfiler Server Tools page
toolsLogedit	edit the View Logs Server Tools page
toolsLogview	access the View Logs Server Tools page
toolsPluginedit	edit the Startable Plugin System Tools page
toolsPluginview	access the Startable Plugin System Tools page
toolsProfileredit	edit the Guidewire Profiler System Tools page
toolsProfilerview	access the Guidewire Profiler System Tools page
toolsWorkQueueedit	edit the Work Queue Info System Tools page
toolsWorkQueueview	access the Work Queue Info System Tools page
viewmednote	view a medical note
viewpolycysystem	view policy in policy system
viewrefdata	view administration reference data
zonemanage	create, edit, or delete administrative zones

Code	Permission to...
zoneview	view the list of administrative zones

The View Report Tab (`reporting_view`) permission in ClaimCenter 6.0 is added to the Adjuster role in the default application. However, the upgrade does not add this permission to the Adjuster role because you might not want that change in your configuration. You can add this permission to the Adjuster role using the **Administration** tab. See “Working with Permissions and Roles” on page 381 in the *Application Guide* for instructions.

ClaimCenter 6.0 adds the following Report Permission Sets:

```
viewaggclaimmetrics
viewownmetricalerts
viewsupmetricalerts
```

See “Administering Reports” on page 105 in the *Reporting Guide*.

Changes in ClaimCenter 6.0

This topic describes what has changed in ClaimCenter 6.0. This includes:

- “Administration Tab Contains New Items” on page 29
- “Advanced Searches” on page 29
- “Business Weeks” on page 30
- “Currency Aware Authority Limit Profile” on page 30
- “FNOL Wizards Are Updated” on page 30
- “Homeowner’s Screens Have Been Expanded” on page 30
- “Integration with Policy Administration Systems” on page 30
- “ISO Expanded” on page 30
- “Workers’ Compensation Screens Have Been Reorganized” on page 30

Administration Tab Contains New Items

ClaimCenter 6.0 either expands or provides additional administrative support for the following functions:

- Catastrophes
- WC Parameters
- Reinsurance Threshold
- ICD Codes
- Metrics and Thresholds
- Business Week

See “The Administration Tab” on page 399 in the *Application Guide*.

Advanced Searches

Use the search tab to perform searches for claims, activities, checks, recoveries, and bulk invoices. For detailed information, see “Performing Searches” on page 49 in the *Application Guide*.

Business Weeks

The `config.xml` file contains business calendar parameters. These parameters are applied system wide and are the default. However, you can configure business weeks to a more granular level which then overrides the configuration parameters. Doing so allows you to define business weeks based on zones.

See “Business Weeks” on page 258 in the *Application Guide*.

Currency Aware Authority Limit Profile

Multicurrency support has been broadened for carriers who write policies in more than one country where the countries have different currencies. Carriers can manage their claims for all of these countries in one instance of ClaimCenter. You do this through the Authority Limit Profile in the **Administration** tab where a currency field has been added.

See “Authority Limit Profile” on page 403 in the *Application Guide*.

FNOL Wizards Are Updated

Certain lines of business: workers’ compensation, homeowner’s, commercial auto, and commercial property have updates to their wizards reflecting a closer match to business requirements.

See “New Claim Wizard and the Lines of Business” on page 68 in the *Application Guide*.

Homeowner’s Screens Have Been Expanded

Contents and scheduled items sections have been added to the **Loss Details** screen.

See “Homeowner’s Line of Business” on page 285 in the *Application Guide*.

Integration with Policy Administration Systems

ClaimCenter has further integrated with policy administration systems (or with PolicyCenter).

- Users can open an instance of the policy system (or PolicyCenter) if it is web based.
- Administrators can set limits of large loss thresholds for a policy system (besides setting those for ClaimCenter).
- See “Policy System Integration” on page 335 in the *Application Guide*.

ISO Expanded

ClaimCenter now supports integration at both the claim messaging level and at the exposure level. If you choose to integrate with ISO, then you must configure ClaimCenter to be integrated either at the exposure *or* the claim level. Depending on that integration, accessing ISO in the user interface can be in several locations.

See “ISO and Claims” on page 361 in the *Application Guide*.

Workers’ Compensation Screens Have Been Reorganized

The screens in the workers’ compensation line of business have been reorganized to provide a more intuitive approach in the location of data. **Time Loss**, **Medical Details**, and **Employer Liability (exposures)** are now accessed from the left pane of the user interface.

See “Workers’ Compensation Screens” on page 298 in the *Application Guide* for details.

Minor Exposure and Typelist Method Changes

A few minor domain methods that were unlikely used by customers changed in exposures. If you used them, you will get compile errors you can easily correct using the instructions below.

Removed `Exposure.isSSBenefits()`

This method converted the `SSBenefit` boolean property into a `String`. It also inverted it, returning "false" if the property was true. This method is now gone. Instead, use the `SSBenefit` property directly. Use "as String" to convert it to a string and of course add the keyword `not` if you need to invert the boolean logic.

Removed `Exposure.isWageBenefits()`

This method converted the `WageBenefit` boolean property into a `String`. It also inverted it, returning "false" if the property was true. This method is now gone. Instead, use the `WageBenefit` property directly. Use "as String" to convert it to a string and of course add the keyword `not` if you need to invert the boolean logic.

Removed `Exposure.isWorkersCompBenefits()`

This method converted the `WCBenefit` boolean property into a `String`. It also inverted it (returning false if the property was true). This method is now gone. Instead, use the `WCBenefit` property directly. Use "as String" to convert it to a string and of course add the keyword `not` if you need to invert the boolean logic.

Removed `libraries.ScreenEditFunctions.SaveATypeListValue(java.lang.String)`

This function returned its argument unchanged. This method is now gone.

New and Changed in Configuration in 6.0

This topic includes:

- “Changes to the Guidewire ClaimCenter Data Model” on page 33
- “Changes in Studio” on page 42
- “Changes Related to PCF Files” on page 44
- “Changes Related to Workflow” on page 44
- “Improvements in Localization” on page 45
- “Catastrophe Bulk Associations Batch Job” on page 48
- “Changes in Duplicate Claim and Check Searches” on page 48
- “Configuring Claim Health Metrics” on page 48
- “Configuring Deductibles” on page 48
- “Configuring Approval Rules with Bulk Invoice Checks” on page 48
- “Configuring Recently Viewed Claims in the Claim Tab” on page 49

Changes to the Guidewire ClaimCenter Data Model

Guidewire has made the following changes to the ClaimCenter 6.0 base data model.

- New Configuration Parameter Access
- Added, Removed, or Changed Configuration Parameters
- Localized Database Columns
- Reference Activity Pattern by Code, Not ID
- New Metadata File Structure

- Changed FieldValidators File
- Changes to Data Types
- Changes to Entity Types
- Changes to Data Objects
- Changes to Database Tables
- Contact Casting Accessors Deprecated
- Column Attributes Become columnParam Subelements
- Columns Added to HolidayZone Entity
- Removed Entity Properties
- Renamed Entity Properties

New Configuration Parameter Access

ClaimCenter 6.0 defines configuration parameters in Java code. (Prior to ClaimCenter 6.0, Guidewire defined configuration parameters through an XML file.) To access a configuration parameter in Gosu code, use the following syntax, with XX being PL, BC, CC, or PC as appropriate.

```
gw.api.system.XXConfigParameters
```

For example:

```
var businessDayEnd = gw.api.system.PLConfigParameters.BusinessDayEnd.Value
var forceUpgrade   = gw.api.system.PLConfigParameters.ForceUpgrade.Value
```

Added, Removed, or Changed Configuration Parameters

ClaimCenter 6.0 adds, removes or changes the following configuration parameters:

Added System Cache Configuration Parameters

- GlobalCacheActiveTimeMinutes
- GlobalCacheReapingTimeMinutes
- GlobalCacheSizeMegabytes
- GlobalCacheSizePercent
- GlobalCacheStaleTimeMinutes
- GlobalCacheStatsRetentionPeriodDays
- GlobalCacheStatsWindowMinutes

Added ClaimCenter Configuration Parameters

- AllowUnsupportedDatabaseVersion
- AllowUnsupportedJDBCDriver
- ArchiveDatabase
- BatchProcessHistoryPurgeDaysOld
- CheckUnableToStopPattern
- CheckUnableToVoidPattern
- ClaimHealthCalcMaxLossDateInYears
- EnableDecentralizedAdmin
- InitialReserveAllowedPeriod
- InstrumentedWorkerInfoPurgeDaysOld
- KeepCompletedMessagesForDays
- LastPaymentReminderPattern
- LegacyExternalEntityArraySupport
- MaxCatastropheClaimFinderSearchResults
- MaxClaimResultsPerClaimHealthCalcBatch
- PolicySystemURL
- ProfilerDataPurgeDaysOld

- QuickJumpShortcut
- ShouldSynchUserRolesInLDAP
- StrictDataTypes
- UseDeductibleHandling
- WizardNextShortcut
- WizardPrevShortcut
- WorkflowLogPurgeDaysOld
- WorkflowPurgeDaysOld
- ZoneCacheRefreshIntervalSecs

Removed ClaimCenter Configuration Parameters

- BaseURLBuilder
- BucketSizeForHistogramsOnAllIndexedColumns
- CollectHistogramsOnAllIndexedColumns
- DatabaseCounterThresholdArchiveSecs
- DisableCBQTFForTeamGroupActivities
- DisableHashJoinForTeamGroupActivities
- DisableIndexFastFullScanForTeamGroupActivities
- DisableSortMergeJoinForTeamGroupActivities
- IncrementalReaderChunkSize
- IncrementalReaderPollIntervalMillis
- IncrementalReaderSafetyMarginMillis
- IntegratedStackTraces
- PCFVerificationMode
- PolicyCenterPassword
- PolicyCenterTimeout
- PolicyCenterURL
- PolicyCenterUser
- PolicySearchDemoAdapterImpl
- ProfilerFlushInterval
- QuickjumpShortcut
- ReportAdminLogin
- ReportManagerURL
- ReportProxyScheme
- ReportProxyServerPort
- ReportProxyURI
- SortWSDLsUponStudioImport
- TextAreaSizeLimit
- UseCityAsCountyForCanadaAssignment
- XMLPolicySearchDir
- XMLPolicySearchFileList

Localized Database Columns

Guidewire has changed the base data model of ClaimCenter 6.0 to support the use of a *localization* column on those entities that ClaimCenter stores in the database. This includes many types of shared administration data such as activity patterns. The end result, using activity patterns as an example, is that you can localize the subject of an activity pattern.

The <localization> subelement on the <column> element takes the following form:

```
<column ... >
  <localization tableName="..."/>
</column>
```

The attribute `tableName` is the table name of the localization join table. ClaimCenter creates this join table automatically if you add a localization column to an entity. You must then modify the appropriate PCF files to display the localization information.

The end result is that you can localize the subject of an activity, for example. You can also create a note, email, or document template with certain fields localized.

See Also

- For details, see “Localizing Shared Administration Data” on page 501 in the *Configuration Guide*.

New Language Column on Document and Activity Entities

In ClaimCenter 6.0, Guidewire has modified the Document and Activity entities by adding a Language column. The default value is `null`. However, because Guidewire only exposes `ILocale` in Gosu, the property displays as `Locale`, rather than `Language`, in Gosu code.

Guidewire has modified the `DocumentSearchCriteria` object accordingly to reflect the changes to the Document and Activity entities.

Reference Activity Pattern by Code, Not ID

With ClaimCenter 6.0, you reference an activity pattern by code value. Prior to ClaimCenter 6.0, you used the activity pattern ID to reference an activity pattern. For example:

```
ActivityPattern( 50/* Review */ )
ActivityPattern( "default_data:12001" /* approval_denied */ )
```

Instead, use the `createActivityFromPattern` method and supply the actual activity pattern code, instead of the activity pattern ID.

For example, use:

```
"approval_denied"
```

Rather than:

```
"default_data:12001"
```

New Metadata File Structure

Guidewire has changed the ClaimCenter 6.0 data model extensively. Prior to ClaimCenter 6.0, the ClaimCenter data model defined multiple entities in a single metadata file named for the primary entity in the file. ClaimCenter also defined all data model extensions in a single `extensions.xml` file.

IMPORTANT Guidewire has removed `extensions.xml`. It no longer exists. Instead, use the new data model files to extend the base configuration data model. You no longer need to increment a version number if you modify the data model either. Guidewire has removed that requirement as well.

ClaimCenter defines each base entity and extensions entity in its own separate metadata file. The file extension of the metadata file depends on the entity type:

Extension	Meaning	Contains	Entity type
.dti	Data Type Info	A single data type definition.	<code>datatype</code>
.eix	Entity Internal eXtension	A single entity internal extension for use by Guidewire only.	<code>internalExtension</code>

Extension	Meaning	Contains	Entity type
.eti	Entity Type Information	A single entity declaration. The name of the file corresponds to the name of the entity.	component delegate deleteEntity entity nonPersistentEntityc subtype viewEntity
.etx	Entity Type eXtension	A single entity extension.	extension viewEntityExtension
.tix	Typelist Internal eXtension	A single typelist extension for use by Guidewire only.	internalTypelistExtension
.tti	Typelist Type Info	A single typelist declaration. The name of the file corresponds to the name of the typelist.	typelist
.ttx	Typelist Type eXtension	A single typelist extension.	typelistExtension

New Way to Create Entity Extensions

To extend an existing data entity or to create a new entity in releases prior to ClaimCenter 6.0, you added entries to file `extensions.xml`. ClaimCenter 6.0 replaces this system with the ability to create a separate extension file for each extended entity (with file extension `.etx`). In doing so, Guidewire has removed file `extensions.xml` totally. It no longer exists.

With ClaimCenter 6.0, to create an extension file, do the following:

1. Select an object file in the **Resources** tree (`Account.eti`, for example).
2. Select **Create extension file** from the right-click submenu.

Studio creates a correctly named extension file (`Account.etx`, in this case), places it in the correct directory, and opens the file in a view tab for editing.

Removal of Version Number

Prior to ClaimCenter 6.0, the entity extension file (`extensions.xml`) contained a version number that you needed to increment each time that you modified the data model. Otherwise, a server restart would throw an error. *Guidewire has removed this file and this requirement.* You no longer need to track or increment a version number as you modify the ClaimCenter data model.

New XML Namespace Attribute

In ClaimCenter 6.0 Guidewire embeds an `xmlns` attribute in all data object XML files in the base configuration. It is:

```
xmlns="http://guidewire.com/datamodel"
```

The purpose of this attribute is to provide a *namespace* for use by XML editors. Its use is optional.

See Also

- “Configuring External Editors” on page 101 in the *Configuration Guide*

Changed FieldValidators File

In ClaimCenter 6.0, Guidewire has split the `fieldvalidators.xml` file into two files. You can find these files in Studio at **Resources** → **Data Model Extensions**.

File	Contains
<code>fieldvalidators.xml</code>	Field validator information only.

File	Contains
<code>datatypes.xml</code>	Abstract data type information only.

Locale-specific field validators. Moving field validator definitions into its own file enables you to create country-specific field validator files, using a file naming convention of `fieldvalidators_<country_code>.xml`. Place any country-specific files that you create in the **Resources** → **Data Model Extensions** → **fieldvalidatorscountries** folder.

IMPORTANT Use the two-character ISO designation for the country code. This is **not** the same as the *locale*, which includes both a country code and a language code.

Validator field overrides obsolete. Guidewire has removed the ability to create `<FieldValidatorOverride>` elements in the `fieldvalidators` file. Prior to ClaimCenter 6.0, you used this element to use to change the default validator used by fields in the application. Starting in ClaimCenter 6.0, you perform this functionality by using the `<column-override>` element in the specific entity extension file.

Validator definitions in `contact-localization-config.xml` obsolete. Guidewire has removed the ability to create `<ValidatorDef>` elements in the `contact-localization-config.xml` file. Starting in ClaimCenter 6.0, you create country-specific field validation files.

Localized field validation error messages. Starting in ClaimCenter 6.0, you can create localized error messages (display keys) and link them to country-specific field validators. You link the display keys to validator definitions contained in the country-specific field validation files.

ClaimCenter checks that field data matches field validator format. Starting in ClaimCenter 6.0, Guidewire ClaimCenter checks that the field data matches the field validator format (the regular expression) as it sets the field on the object. Thus, you cannot, for example, assign a value to a **ClaimNumber** field that does not match the acceptable **ClaimNumber** format as defined for this field in `fieldvalidators.xml`.

This is a change from releases prior to 6.0. If you want to preserve the previous behavior, you can specify this by setting the following configuration parameter in `config.xml`.

```
StrictDataTypes=false
```

See Also

- “Localizing Field Validators” on page 507 in the *Configuration Guide*

Changes to Data Types

Guidewire has made the following changes to the data types in ClaimCenter 6.0.

Column types become data types. Prior to ClaimCenter 6.0, the ClaimCenter data model defined a set of enumerated column types (`datetime` or `decimal`, for example). Thus, if you defined a new column on an entity, you needed to set the column type as well. With the release of ClaimCenter 6.0, the `type` attribute on the `<column>` element is a string and must be the name of a valid data type. All the column types become data types as of the release of 6.0.

New ability to create data types. Prior to ClaimCenter 6.0, it was not possible for you to define your own data types. Guidewire has removed this restriction on the data model. Starting in ClaimCenter 6.0, you can define your own data types. See “Data Types” on page 299 in the *Configuration Guide* for details.

New Money data type attribute. In ClaimCenter 6.0 Guidewire has added a new attribute to the **Money** data type called `appscale`. This attribute controls the number of digits shown within ClaimCenter for money or currency. This attribute functions in a similar manner to the `scale` attribute, but overrides it if it is present. See “Data Types” on page 299 in the *Configuration Guide* for details.

New currency-related data types. Guidewire has added the following currency-related data type to Guidewire ClaimCenter 6.0:

CurrencyAmount	Permits positive, negative, and zero values.
NonnegativeCurrencyAmount	Does not permit negative values. However, zero values are acceptable.
PositiveCurrencyAmount	Does not permit negative or zero values.

If you create a custom entity with `XXCurrencyAmount` as the data type, then you need to define a `getXXCurrency` method on the property in order to retrieve the currency amount. For example:

```
<column name="SomeAmount" type="currencyamount" ...>
  <columnParam name="currencyProperty" value="ClaimCurrency"/>
</column>
```

Changes to Entity Types

Guidewire has made the following changes to the base configuration entity types in ClaimCenter 6.0:

Entity	Modification
delegate	It is possible to define your own delegates and to extend the base configuration delegates. See "Creating a New Delegate Object" on page 241 in the <i>Configuration Guide</i> for details.
delegate edgeForeignKey entity extension subtype	Guidewire has added an <code><implementsEntity></code> subelement to the XML definitions of these entity types. You use this element to define delegate objects. See " <code><implementsEntity></code> " on page 224 in the <i>Configuration Guide</i> for details.
delegate entity extension subtype	Guidewire has added an <code><implementsInterface></code> subelement to the XML definitions of these entity types. You use this element to indicate that this entity implements the specified interface. See " <code><implementsInterface></code> " on page 225 in the <i>Configuration Guide</i> for details.
nonPersistentEntity	It is possible to create new <code>nonPersistentEntity</code> objects in the extensions folder.
viewEntity	It is possible to create <code>viewEntity</code> data objects. In the past, you could merely extend (subtype) existing <code>viewEntity</code> objects. See "View Entity Data Objects" on page 207.

Changes to Data Objects

Guidewire has made the following changes to the following data model data objects in ClaimCenter 6.0:

Element	Attribute	Modification
array foreignkey typekey	immutable	Guidewire has removed the <code>immutable</code> attribute (column) on the array, foreignkey, and typekey objects.
column	supportsCaseInsensitiveSearch	Guidewire has renamed this attribute to <code>supportsLinguisticSearch</code> . For more details, see "Localized Search and Sort" on page 525.
column edgeForeignKey foreignkey typekey	createhistogram	It is possible to change the value of the <code>createhistogram</code> attribute through the use of the <code><column-override></code> parameter in an extension file. See "Working with Attribute Overrides" on page 239 in the <i>Configuration Guide</i> for details.
foreignkey	createbackingindex	Guidewire has added a new attribute <code>createbackingindex</code> to <code><foreignkey></code> . If true, the database automatically generates a backing index for the foreign key. See " <code><foreignkey></code> " on page 220 in the <i>Configuration Guide</i> for details.

Changes to Database Tables

Guidewire has modified the following database tables in ClaimCenter 6.0. They are no longer retirable. These tables no longer have a corresponding staging table.

- cc_claimmetric
- cc_exposuremetric
- cc_claimmetricrecalcTime

Removal of <validationTriggerOverrides> Element

You no longer use the <validationTriggerOverrides> element (in extensions.xml) to trigger validation upon modification of an entity in the ClaimCenter base data model that does not currently trigger validation. Neither the <validationTriggerOverrides> element or the extensions.xml file exist as of ClaimCenter 6.0.

Instead, use a triggersValidation attribute on an extension entity to modify this behavior. See “Working with Attribute Overrides” on page 239 in the *Configuration Guide* for more information.

Contact Casting Accessors Deprecated

With ClaimCenter 6.0, Guidewire has deprecated all the *casting accessors* on the Contact object (for example, Contact.Person, Contact.Company, and similar items). If you attempt to use one of these accessors, ClaimCenter reports a warning message, such as:

Person in entity.Contact has been deprecated.

Instead, use the following syntax:

```
(obj typeis Person ? obj : null)
```

For example, if contact is a variable of type Contact, you can do the following:

```
(contact typeis Person ? contact : null)
```

You can encounter this situation frequently, for example, in PCF pages. If so, Guidewire recommends that you declare a variable of the type you want and perform the cast as you call the PCF page. For example, declare a variable contactPerson of type Person.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Review PCF pages, libraries and rules for deprecated usages of Contact.Person and similar casting accessors on the Contact object. Update as necessary.

Column Attributes Become columnParam Subelements

For ClaimCenter 6.0, Guidewire has modified the data model so that attributes of <column> entities are separate <columnParam> elements. The following former <column> attributes have become <columnParam> attributes:

- size
- scale
- precision
- encryption
- trimwhitespace

For example, the entity:

```
<entity entity="TestOverride" ...>
  <column name="ScaleOverride" precision="5" scale="2" type="decimal"/>
</entity>
```

becomes:

```
<entity entity="TestOverride" ...>
  <column name="ScaleOverride" type="decimal"/>
```



```
<columnParam name="precision" value="5"/>
<columnParam name="scale" value="2"/>
</column>
</entity>
```

Starting in ClaimCenter 6.0, Guidewire makes a distinction between the `size` attribute and the `logicalSize` attribute.

- `varchar`-based data types use the `size` attribute. You use it to specify the size of the column in the database.
- `String`-based data types use the `logicalSize` attribute. You use it to add an ClaimCenter-enforced constraint on the size of the data.

Thus, you can define a column with `size` of 50 in the database. However, if column has a `logicalSize` of 30, then ClaimCenter only allows strings of length 30 or less.

In ClaimCenter 6.0, if the column type was `encryption`, the column type becomes `varchar` and the column has the following `<columnParam>` element:

```
<columnParam name="encryption" value="true"/>
```

Columns Added to HolidayZone Entity

For ClaimCenter 6.0, Guidewire has added new columns to the `HolidayZone` entity, the entity that associates the `Holiday` object with `Zone` object. The new columns are:

- `ZoneType`
- `Country`

Together with `Code`, the `Code-ZoneType-Country` triplet uniquely identifies a zone. This was not the case in releases prior to ClaimCenter 6.0. In releases prior to ClaimCenter 6.0, multiples zones could have identical codes. For example, the code `CA` could mean both Canada and California. The code `New York` could mean both the city and the state.

IMPORTANT ClaimCenter automatically updates the database during upgrade to account for the new columns. However, there may be `HolidayZone` rows that have the value `unknown` in those new columns. You must change these values manually to their correct values.

UPGRADE TASK

Priority: Before you start the production server.

Summary: Review table `HolidayZone` table for empty `ZoneType` and `Country` columns. Update as necessary.

Removed Entity Properties

Guidewire has removed the following entity properties from the `Claim` entity:

- `DeductibleStatus`

Guidewire has removed the following entity properties from the `Transaction` entity:

- `Netamount`
- `OriginalAmount`
- `DeductibleAmount`
- `DeductibleSubtracted`
- `DeductiblePaid`

See Also

- “Removed Rules” on page 111 for a list of business rules that Guidewire has removed because the rules reference these removed properties.

Renamed Entity Properties

Guidewire has renamed the following entity property names:

Property name	New property name
Deduction.Amount	Deduction.ClaimAmount
CheckPortion.FixedAmount	CheckPortion.FixedTransactionAmount

Property getter and setter methods continue to exist for the old property names. However, in ClaimCenter 6.0, these are *virtual* properties and they defer to the new getter and setter methods internally. This change does not typically affect existing installations as existing Gosu code can continue to refer to the old property names.

There are certain types of access that Guidewire does not permit with virtual properties. Thus, you need to update the following types of usages for the listed properties:

- References to `Deduction.Amount` or `CheckPortion.FixedAmount` from Gosu *find* expressions
- String references to the old property names; for example, if used in calls to `Bean.ifFieldChanged(String)`

UPGRADE TASK

Priority: Before starting the server

Summary: Update references to old property names

Changes in Studio

In ClaimCenter 6.0, Guidewire has made the following changes to Guidewire Studio:

- Changes to Studio Dialogs
- Changes to Studio Editors
- Changes to Studio Menus
- Changes to Studio Right-Click Contextual Menus

In addition to these changes, ClaimCenter 6.0 provides the ability to set a locale for the Studio interface. See “Set a Locale for Studio” on page 45 for more information.

Changes to Studio Dialogs

Guidewire has made the following changes to the Studio dialogs:

Dialog	Change
Create Display Key	Guidewire added a new Specify values for each locale checkbox to the Create Display Key dialog. Check this box if you want to create a localized version of a display key. See “Different Ways to Localize Display Keys” on page 485 in the <i>Configuration Guide</i> for details.
Verify	<p>Guidewire modified the Studio Verify dialog in the following ways:</p> <ul style="list-style-type: none"> • Guidewire includes all the children of a resource in the verification process for that resource. The tool does this automatically, unless otherwise specified. As such, the Verify dialog no longer contains an Verify children option (checkbox). This is also true of the Verify dialog that you reach by selecting Find in path option from the right-click menu. • Guidewire labels the Incremental verify verification option Verify changed resources instead. The functionality remains the same. <p>For information on the verification tool, see “Validating Rules and Gosu Code” on page 173 in the <i>Configuration Guide</i>.</p>

Changes to Studio Editors

Guidewire has made the following changes to the Studio editors in Guidewire 6.0:

Editor	Change
Gosu Templates	Guidewire has added the ability to create and manage Gosu templates through Studio. You can create a Gosu template in the Classes folder by selecting a package and then selecting New → Template from the right-click menu.
Guidewire XML Model	Guidewire has added the ability to export business data entities (and other types such as Gosu classes) to XML. In Studio, navigate in the resource tree to the package hierarchy in which you want to store your XML model. Next, right-click on the package and from the right-click menu choose New → Guidewire XML Model . For more information, see “The Guidewire XML (GX) Modeler” on page 262 in the <i>Gosu Reference Guide</i> .
PCF	Guidewire added a new Translations tab on the right-hand side of the editor. This tab displays a list of all display keys used in the PCF file. Double-clicking a display key opens an Edit Display Key dialog that shows the value of the display key in each locale defined within ClaimCenter. You can use this dialog to create locale-specific display keys that are specific to the context in which they exist. See also “The Translations Tab” on page 340 in the <i>Configuration Guide</i> .
Rules	Guidewire has made the following changes to the Rules editor: <ul style="list-style-type: none"> Guidewire has removed the list of changed or modified files that existed at the bottom of the middle pane in the Rules editor. Guidewire has added the ability to create rule set categories and rule sets through Studio.
Workflow	Guidewire has added the ability to create the necessary metadata for a new workflow subtype through Studio. To access this functionality, select Workflows in the Resources tree, then Create metadata for new workflow subtype from the right-click contextual menu.

Changes to Studio Menus

Guidewire has made the following changes to the Studio menus:

Menu	Change
Debug	Guidewire has merged the Debug menu commands of Debug → Rules and Debug → Web into a single Debug → Server command. Use this command if you are attempting to debug Gosu code running on a connected server.
Help	Guidewire added several new menu items to the Studio Help menu. These include: <ul style="list-style-type: none"> Gosu Reference Guide, which opens the Guidewire ClaimCenter documentation suite in a separate browser window. PCF Reference Guide, which opens the <i>Guidewire ClaimCenter PCF Format Reference</i> in a separate browser window.
GUnit	Guidewire has moved the GUnit Run and Debug commands to their own separate GUnit menu (from the Debug menu). Other than this change, the GUnit functionality remains the same.
Tools	Guidewire has removed the Rule Repository Report command (and its associated functionality) from Guidewire ClaimCenter.

Changes to Studio Right-Click Contextual Menus

Guidewire has made several additions to the contextual (right-click) menu, all of which provide new functionality such as the ability to create new types of resources through Studio. The exact menu you see depends on the context, the resource that you select. The new menu items include:

- Classes** → *package* → **New** → **Template**
- Classes** → *package* → **New** → **Guidewire XML Model**
- Data Model Extensions** → *extensions* → **.eti* → **Create extension file**
- Other Resources** → *datatypes* → **New** → **Other file** → **.dti*
- Rule Sets** → **New** → **Rule Set Category**

- Rule Sets → *rule set category* → New → Rule Set
- Workflows → Create metadata for new workflow subtype

Changes Related to PCF Files

In ClaimCenter 6.0, Guidewire has made the following changes related to PCF files:

- PCF Schema Changes
- Server Start Up No Longer Verifies PCF Files

PCF Schema Changes

Removed properties. Guidewire has removed properties from the following widgets:

Widget	Removed properties
DateInput	inputConversion outputConversion
LocationEntryPoint	parent canVisit
ProgressCellNode	sortBy sortDirection sortOrder
TypeKeyInput	outputConversion

Nested wizard steps. Guidewire no longer permits you to create nested wizard steps. An upgrade process automatically converts nested wizard steps into a wizard step group at the same level as the parent step.

Search Panel widget. Guidewire no longer permits you to create a PCF page using a Search Panel widget as the top-level element. You can, however, still use this widget as a lower-level element.

Server Start Up No Longer Verifies PCF Files

Guidewire has removed PCF verification at server start up. In its place, Guidewire has added a *verify-all* procedure to the `build-war` task.

- To verify PCF files, use the verification tools in Guidewire Studio.
- To reload PCF files, use the **Verify All PCF Files** button on the **Internal Tools** page, or the ALT-SHIFT-L shortcut from within the application. This operation reloads the PCF files but does not verify them. You need to use Guidewire Studio to perform PCF verification.

IMPORTANT Guidewire has also removed the `PCFVerificationMode` configuration parameter.

Changes Related to Workflow

In ClaimCenter 6.0 Guidewire has made a number of changes to workflow. These include:

- Studio provides a right-click menu command that you can use to change the step type within a workflow.
- Studio provides the ability to create the necessary metadata for a new workflow subtype.
- Studio opens the workflow to proper workflow step if you use the **Find in path** command.

See Also

- For general information on Guidewire workflow, see “Guidewire Workflow” on page 423 in the *Configuration Guide*.

Improvements in Localization

Guidewire has made the following changes to Guidewire ClaimCenter 6.0 to improve and simplify localization of the product.

- Set a Locale for Studio
- Localize Specific Activity Pattern Fields
- Localize Field Validation
- Localize Documents, Notes, and Templates
- Japanese Imperial Calendar Date Picker
- Localization of Workflow
- Localization of Command Line Tools Argument Descriptions

Set a Locale for Studio

If you configure the application for multiple locales properly, Studio displays a drop-down locale selector at the bottom right-hand corner of the screen. You can use this selector to set the Studio locale to one of the defined application locales. However, this has no meaning unless you have defined certain property files for that localization. If you define the necessary localized property files, Studio provides localized versions of the following:

- Studio field labels
- Rule set names and descriptions
- Gosu error messages
- Workflow names and step names

See Also

- “Localizing the Development Environment” on page 491 in the *Configuration Guide*

Localize Specific Activity Pattern Fields

ClaimCenter 6.0 includes the ability to generate localized versions of certain activity pattern properties (or fields) such as the activity pattern subject or description. If you configure ClaimCenter correctly to use multiple locales, then you see additional fields at the bottom of the **New Activity Pattern** screen. You use these fields to enter localized subject and description text for that activity pattern. You can create additional localized fields by extending the data model and creating additional localized columns on the `ActivityPattern` object.

Localize Field Validation

ClaimCenter 6.0 provides the ability to create localized field validation by creating country-specific field validator files. See “Localizing Field Validators” on page 507 in the *Configuration Guide* for details.

Set Zone Information

ClaimCenter 6.0 provides the ability to define specific zone information to use for region and address auto-fill in file `zone-config.xml`. (You access this file in Studio in the **Other Resources** folder.) See “Configuring Zone Information” on page 476 in the *Configuration Guide* for details.

Localize Documents, Notes, and Templates

ClaimCenter 6.0 provides the ability to create localized versions of the templates used to create new documents, emails, and notes. The process is straight-forward and relatively simple. It mainly involves creating locale-specific folders in the correct location and populating each folder with translated versions of the required document, email, or note templates. If you enable localization, ClaimCenter automatically recognizes the presence of localized templates in these folders. A user can then select a localized template as the basis of a new note, document, or email.

Localize Document APIs

Guidewire has moved all document plugin interface definitions to the following package:

```
gw.plugin.document
```

Guidewire has moved all implementations of these interfaces to the following package:

```
gw.plugin.document.impl
```

Guidewire has moved methods to handle document creation to the following package:

```
gw.document
```

IDocumentTemplateDescriptor

Guidewire has modified the IDocumentTemplateDescriptor interface so that it presents a template in a certain locale. The interface contains several additional methods:

- `document.IDocumentTemplateDescriptor.getName(locale)`
- `document.IDocumentTemplateDescriptor.getLocale()`

IDocumentTemplateSource

Guidewire has modified the IDocumentTemplateSource interface by deprecating several methods and creating new methods with the same name but with `locale` as an additional parameter.

- `getDocumentTemplate(date, valuesToMatch, maxResults)`
- `getDocumentTemplate(templateId, locale)`
- `getTemplateAsStream(templateId, locale)`

If you need to search on a template locale, you can supply the locale in the `valuesToMatch` parameter.

IDocumentTemplateSerializer

Guidewire has added the following method to the IDocumentTemplateSerializer interface.

```
document.IDocumentTemplateSerializer.localize(locale, descriptor)
```

This method converts a template descriptor to the given locale.

DocumentTemplateSearchResultBase

Guidewire has deprecated the DocumentTemplateSearchResult class. Starting with ClaimCenter 6.0, the `performSearch` method—on DocumentTemplateSearchCriteria—returns an array of IDocumentTemplateDescriptor objects from which you can determine the locale.

Document File Structure Changes

To make document templates localizable, Guidewire has modified the directory structure to handle multiple, localized, templates. With ClaimCenter 6.0, each template descriptor can have multiple templates, one for each supported locale, and one for the default. In this way, if you do not choose to support localization, you can maintain your current configuration without a need to upgrade the file structure.

The directory structure looks similar to the following (for example):

```
modules/configuration/config/resources/doctemplates
+-- EmailSent.gsceipt.htm                //template file
```

```

+-- EmailSent.gscript.htm.descriptor      //template descriptor
+-- ...                                  //other templates and template descriptors
+-- en_US
|   +-- EmailSent.gscript.htm             //template file in English
|   +-- ...                               //other template files in English
|-- fr_CA
|   +-- EmailSent.gscript.htm             //template file in French
|   +-- ...                               //other template files in French
...

```

As a consequence, you can search for a document template that supports a specific locale in the ClaimCenter interface and use it to create a localized document. Thus, it is possible for someone working in one locale to create a document localized to another locale.

Localized Field Validation

See “Changed FieldValidators File” on page 37 for details.

Japanese Imperial Calendar Date Picker

ClaimCenter 6.0 added support for the use of the Japanese Imperial calendar date picker. You can toggle back and forth between the Gregorian calendar and the Japanese Imperial calendar in any date picker if you enable this functionality. You can set a default calendar for each locale through the use of the `defaultCalendar` attribute on the `<GWLocale>` element in the `localization.xml` file. See “Working with the Japanese Imperial Calendar” on page 541 in the *Configuration Guide* for more information.

Localization of Command Line Tools Argument Descriptions

ClaimCenter 6.0 added the ability to localize the descriptions of the command line tool arguments. For more information, see “Localizing Administration Tool Argument Descriptions” on page 494 in the *Configuration Guide*.

Localization of Workflow

At the start of the workflow execution, the Workflow engine evaluates the workflow locale and uses that locale for notes, documents, templates, display keys, dates, numbers, and other similar items. Unless set otherwise, this is the application default locale.

ClaimCenter 6.0 added the following localization capabilities to workflow:

- The ability to set a locale for a workflow that is different from the default application locale.
- The ability to create localized workflow logs that you view in the ClaimCenter **Administration** tab.
- The ability to view localized workflow step names in Studio.
- The ability to add localized Gosu blocks (meaning blocks of Gosu code that run in a specific locale) to workflow steps. (Indeed, you can run localized Gosu blocks of code in any Gosu code.)

See Also

- For general information on workflow and its improvements, see “Guidewire Workflow” on page 423 in the *Configuration Guide*.
- For information on the ability to localize aspects of Guidewire workflow, see “Localizing Guidewire Workflow” on page 497 in the *Configuration Guide*.

Catastrophe Bulk Associations Batch Job

You can configure the *Catastrophe Bulk Associations* batch job. The purpose of the batch job is to find claims that have not yet been associated to a catastrophe. After the batch job finds a non-associated claim, it creates an activity and assigns it to someone that can determine whether to associate a catastrophe with the claim.

See Also

- “Working with Catastrophe Bulk Associations” on page 639 in the *Configuration Guide*

Changes in Duplicate Claim and Check Searches

You can configure the templates in Gosu so that you can modify the search criteria for duplicate claims and checks. ClaimCenter checks if there are any matching claims or checks to avoid duplication.

See Also

- “Duplicate Check Search” on page 643 in the *Configuration Guide*.

Configuring Claim Health Metrics

You can configure *Claim Health Metrics*. Specifically, you can add a new tier, a high-risk indicator, or a new claim metric.

See Also

- “Configuring Claim Health Metrics” on page 645 in the *Configuration Guide*.
- “Claim Performance Monitoring” on page 315 in the *Application Guide* to learn about this feature.
- “Metrics and Thresholds” on page 418 in the *Application Guide* to learn how to administer claim health metrics.

Configuring Deductibles

ClaimCenter 6.0 supports deductible handling. The “Configuring Deductibles” topic explains how deductibles are structured, and how they interact with checks.

See Also

- “Configuring Deductibles” on page 633 in the *Configuration Guide*
- “Deductible Handling” on page 183 in the *Application Guide*
- metrics.

Configuring Approval Rules with Bulk Invoice Checks

BulkInvoiceItem objects are now processed and their Check objects are created by a distributed work queue. In the base configuration of ClaimCenter, the workers run as sys user, the system user. The result of this change is that for bulk invoice checks, Check.CreateUser is the system user rather than the user that requested the bulk invoice.

Instead of referencing `Check.CreateUser`, approval rules reference the `CheckSet.RequestingUser` or `BulkInvoice.RequestingUser`. This field is set to the current user when the approval ruleset is run. Bulk invoice checks inherit their `RequestingUser` from the `BulkInvoice`.

See Also

- “BulkInvoice Approval” on page 52 in the *Rules Guide*

Configuring Recently Viewed Claims in the Claim Tab

You can configure recently viewed claim information in the **Claim** tab. You use this tab to either create a new claim, search for a specific claim, or access a recently viewed claim from a list. This topic explains how to add the loss date in the recently viewed claims for the auto loss type using Guidewire Studio.

See Also

- “Configuring Recently Viewed Claims” on page 655 in the *Configuration Guide*

New and Changed in Gosu in 6.0

New in Gosu in 6.0

This topic describes new features in Gosu for ClaimCenter 6.0. These include:

- “Gosu Shell” on page 52
- “Gosu Generated Documentation from Type System” on page 52
- “Run Local Command-line Commands from Gosu” on page 52
- “Shell-related APIs” on page 52
- “Increment and Decrement Operators” on page 52
- “New Compound Assignment Operators” on page 53
- “Gosu Templates” on page 54
- “Gosu Composition Syntax” on page 55
- “Object Lifecycle Management with the ‘using’ Keyword” on page 56
- “Type Inference Downcasting” on page 56
- “Profiler Tags” on page 57
- “The Object Equality Operator (===)” on page 57
- “Object Initializer Syntax During Object Creation” on page 58
- “Type System Reflection New APIs” on page 59
- “New Stream Utilities” on page 59
- “Concurrency Utilities” on page 59
- “Checksum APIs” on page 59
- “Run With New Bundle With a User” on page 59

Gosu Shell

With ClaimCenter 6.0 you can run Gosu programs outside the ClaimCenter server using the new Gosu shell. The Gosu shell is the Gosu language engine running in its own process. You can run Gosu programs directly from the Windows command line. Guidewire has created a new file extension for Gosu files intended to be run from the Gosu shell: `.gsp`. For more information, refer to “Running Local Shell Commands” on page 327 in the *Gosu Reference Guide*.

Gosu Generated Documentation from Type System

Integration programmers might want to use the new Gosu documentation that you can generate from the command line using the `gwcc` tool:

```
gwcc regen-gosudoc
```

You will then find the documentation at `ClaimCenter/build/gosudoc/index.html`.

This documentation is particularly valuable for integration programmers implementing plugins in Gosu. The information in the Javadoc-formatted files are more Gosu-like than using the Java generated Javadoc to understand the plugin interfaces. And this documentation includes more hyperlinks between objects than using the Gosu API Reference from within Studio.

For more information about documentation, see “Gosu Generated Documentation” on page 35 in the *Gosu Reference Guide* and “Integration Documentation Overview” on page 18 in the *Integration Guide*.

Run Local Command-line Commands from Gosu

Gosu includes a class called `gw.util.Shell`. It provides methods to run local command-line programs from Gosu. For more information, see “Running Command Line Tools from Gosu” on page 327 in the *Gosu Reference Guide*.

Shell-related APIs

The new `gw.util.Shell` class has other useful APIs:

- You can easily determine whether the host system is Microsoft Windows.
- You can prompt a user for a question, read a line from standard in, and return the result as a `String`

For details, see “Helpful APIs for Command Line Gosu Programs” on page 326 in the *Gosu Reference Guide*.

Increment and Decrement Operators

Gosu now supports all operators in the Java language, including bit-oriented operators. Additionally, Gosu has compound operators such as:

- `++`, which is the increment-by-one operator, supported only after the variable name
- `+=`, which is the increment-and-assign operator, supported only after the variable name

For example, to increment the variable `i` by 1:

```
i++
```

It is important to note that these operators always form *statements*, not *expressions*.

This means that the following Gosu is valid

```
var i = 1
while(i < 10) {
    i++
    print( i )
}
```

However, the following Gosu is invalid because statements are impermissible in an *expression*, which Gosu requires in a `while` statement:

```
var i = 1
while(i++ < 10) { // Compilation error!
    print( i )
}
```

Gosu supports the increment and decrement operator only **after** a variable, **not before** a variable. In other words, `i++` is valid but `++i` is invalid. The `++i` form exists in other languages to support expressions in which the result is an expression that you pass to another statement or expression. As mentioned earlier, in Gosu these operators do not form an expression. Thus you cannot use increment or decrement in `while` declarations, `if` declarations, and `for` declarations. Because the `++i` style exists in other languages to support forms that are *unsupported* in Gosu, Gosu does not support the `++i` form of this operator.

IMPORTANT Gosu now supports the `++` operator after a variable, such as `i++`. Using it before the variable, such as `++i` is unsupported and generates compiler errors.

New Expression Operators

There are new related operators for use in Gosu expressions:

- Bitwise AND (&)
- Bitwise Inclusive OR (|)
- Bitwise Exclusive OR (^)
- Bitwise Left Shift (<<)
- Bitwise Right Shift and Preserve Sign (>>)
- Bitwise Right Shift and Clear Sign (>>>)

For more details, see “Arithmetic Expressions” on page 63 in the *Gosu Reference Guide*.

These new expressions also have related compound assignment operators. See “Variable Assignment” on page 82 in the *Gosu Reference Guide*.

New Compound Assignment Operators

Gosu now supports the following compound assignment operators for use in Gosu statements:

- Add (+=)
- Subtract (-=)
- Divide (/=)
- Multiply (*=)
- Modulo (%=)
- Bitwise AND (&=)
- Bitwise Inclusive OR (|=)
- Bitwise Exclusive OR (^=)
- Bitwise Left Shift (<<=)
- Bitwise Right Shift and Preserve Sign (>>=)
- Bitwise Right Shift and Clear Sign (>>>=)

These operators perform an action on the contents of a variable and then set the variable to the result. For example, the `+=` operator adds a value to the current value in a variable:

```
var i = 10
i += 2
print(i) // prints 12
```

For details and examples of each operator, see “Variable Assignment” on page 82 in the *Gosu Reference Guide*.

Gosu Templates

Templates in Gosu have several new features. You can now include template syntax directly inside of `String` literals. You can now use an easy-to-read new syntax using the dollar symbol followed by a Gosu expression in curly braces:

```
var name = "Gosu"
var str = "I like to program in ${name}."
```

In many cases, this syntax makes `String` concatenation using the `'+'` operator obsolete. In other words, the code above replaces the harder-to-read code:

```
var name = "Gosu"
var str = "I like to program in " + name + "."
```

As in the previous release, you can define templates as self-contained separate files. However, there are new features of template files, which now use the `.gst` (not `.gs`) suffix. New features of Gosu templates:

- Gosu supports type-safe rendering of Gosu templates (`.gst` files). For example, Gosu can check at compile time that embedded Gosu uses correct types, including a new system for passing parameters to a template. Template files that Gosu checks at compile time become first class objects in Gosu that you evaluate at run time.
- More intuitive invocation and package hierarchies for templates
- More intuitive syntax for rendering your template.
- More intuitive syntax for passing parameters.
- New template comments support.
- New feature to extend a template from a class, to simplify calling static methods on a class.

For more details, see “Gosu Templates” on page 291 in the *Gosu Reference Guide*.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Review your code for any `String` literals and Gosu templates that contain `${...}` syntax. If you find any, refactor to avoid accidental use of new Gosu `String` templates, and test your new code.

Review your code for in-line `String` literals that contain text that includes `${...}` syntax. If you find any, refactor to avoid accidental use of new Gosu `String` templates.

Legacy Gosu Templates Deprecated (‘.gs’ Template Files)

The ClaimCenter documentation now primarily describes the new implementation of Gosu templates in `.gst` files. An older version of Gosu template files existed in previous releases. There are features in the new templates that are unavailable in legacy templates. The legacy Gosu templates have the file extension `.gs` rather than `.gst`. The legacy Gosu template format is now deprecated. Do **not** use the legacy template file style for new

templates. For more information about the differences in this format, see earlier in the topic “Gosu Templates” on page 54.

IMPORTANT In the built-in implementation, there are some legacy `.gs` files that did not yet convert to `.gst` files. However, to create new templates, always use the new style not the legacy template style. If you have any of your own legacy templates, start to convert these to the new format. However, the legacy style continues to work in this release.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Convert all Gosu template code to use the new template system. If you need to run templates from web services, write custom web services unique to each integration point.

This upgrade step has the following steps:

1. Review all existing Gosu template code.
2. Convert any use of `.gs` templates to `.gst` templates.
3. Add template variables to all templates as appropriate.
4. Convert all template execution code to use the new APIs.

Name Change is Different for Plugin Templates

The file name suffix for plugin templates is now `.gsm`, not `.gs` or `.gst`. For details, see “Plugin Template Suffix is Now ‘.gsm’” on page 105.

SOAP APIs for Legacy Gosu Templates Deprecated

Because old style `.gs` Gosu templates are now deprecated, the SOAP APIs that relied on them are now also deprecated.

This affects all methods in the ClaimCenter web service interface `IDataExtractionAPI`.

UPGRADE TASK

Priority: Before your next upgrade

Summary: If you need to run templates from web services, write custom web services unique to each integration point.

Convert any use of built-in SOAP APIs for template extraction into custom web services unique to each integration point.

Gosu Composition Syntax

Gosu classes now support a composition syntax. This allows a Gosu class to delegate an interface to another class. In other words, another class implements the interface methods and your class calls the delegate class to handle those methods. You define the delegate in the class definition and reference the delegate through an internal class field. This compositional model allows for easy implementation of objects that are proxies for other objects.

For example, the following Gosu class `MyStringList` defines a delegate for all methods on `List<String>`. That means that other code can interact with a `MyStringList` class as if it directly handles the methods. However, this class delegates the details to another object, in this case an instance of `ArrayList<String>`.

```
class MyStringList implements List<String>
{
```

```
private delegate _internalList represents List<String> = new ArrayList<String>()
}
```

Other code can now use an instance of `MyStringList` and call `ArrayList` methods that the delegate handles.

```
var x = new MyStringList()
x.add( "TestString" ) // this is a method on ArrayList, but we call the method on our custom class
```

For more information, see “Composition” on page 195 in the *Gosu Reference Guide*.

Object Lifecycle Management with the ‘using’ Keyword

If you have an object with a lifecycle of a finite extent of code, you can simplify your code with the new `using` statement. For example, the typical lifecycle of an output stream is to open the stream, use the stream, and then dispose of the stream. If something goes wrong while using the output stream, your code must still close the output stream.

You can simplify your code using the `using` statement as follows:

```
using( var os = SetupMyOutputStream() ) {
    //do something with the output stream
} // Gosu disposes of the stream after it completes or if there is an exception
```

The `using` statement is a more compact and foolproof way of working with these types of resources. The `using` statement is also useful for synchronization, locking and many other things.

For more information, see “Object Lifecycle Management (‘using’ Clauses)” on page 96 in the *Gosu Reference Guide*.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Review your code related to resource management, streams, and locking. Look for places in which the new using clauses might make code easier to understand or safer.

Type Inference Downcasting

To improve the 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. Within the Gosu code bounded by the `if` statement, you do not need to do casting (“as *TYPE*” expressions) to that subtype. Because Gosu confirms that the object has the more specific subtype, Gosu implicitly considers that variable’s type to be the **subtype**, at least within that block of code.

The structure of this type looks like the following:

```
var VARIABLE_NAME : TYPE_NAME

if (VARIABLE_NAME typeis SUBTYPE_NAME) {
    // use the VARIABLE_NAME as SUBTYPE_NAME without casting
    // This assumes SUBTYPE_NAME is a subtype of TYPE_NAME
}
```

For example, the following example shows a variable declared as an `Object`, but downcasted to `String` within an `if` statement block.

Because of downcasting, the following code is valid:

```
var x : Object = "nice"
var strlen = 0

if( x typeis String ) {
    strlen = x.length
}
```

This works because the `typeis` inference is effective immediately and propagates to adjacent expressions.

It is important to note that `length` is a property on `String`, not `Object`. The downcasting from `Object` to `String` means that you do not need an additional casting around the variable `x`. In other words, the following code is equivalent but has an **unnecessary** cast:

```
var x : Object = "nice"
var strlen = 0

if( x typeis String ) {
    strlen = (x as String).length // "length" is a property on String, not Object
}
```

Use automatic downcasting to write easy-to-read and concise Gosu code. Do not write Gosu code with unnecessary casts.

This also works for switch statements that switch on variables declared as `Object`. For example:

```
uses java.util.Date

var x : Object = "neat"
switch( typeof( x ) ){
    case String :
        print( x.charAt( 0 ) ) // without downcasting, this method call would not work without casting
        break
    case Date :
        print( x.Time ) // without downcasting, this property access would not work without casting
        break
}
```

Downcasting also works for ternary conditional expression, such as the following:

```
x typeis String ? x.length : 0
```

Downcasting only happens in the part of the expression that corresponds to it being true (the first part).

For more information, see “Automatic Downcasting for ‘typeis’ and ‘typeof’” on page 302 in the *Gosu Reference Guide*.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Review your code to remove unnecessary downcasting. This makes your Gosu code easier to read and maintain.

Profiler Tags

You can programmatically add hints to your code so that you can time how long different parts of your code take, all integrated into the main profiler user interface.

```
using( var tag = new ProfilerTag("ExecuteDatabaseQuery","Executes a database query") ) {
    tag.setCounterValue( "test", 3 ) // call methods on the profiler tag
    // do your main code here
}
```

For more information about the `using` keyword, see “Object Lifecycle Management (‘using’ Clauses)” on page 96 in the *Gosu Reference Guide*.

The Object Equality Operator (===)

In the Java language, the `==` operator evaluates to true if and only if both operands have the same exact **reference value**. In other words, it evaluates to true if they refer to the same object in memory. This works well for primitive types like integers. For reference types, this usually is not what you want to compare. Instead, to compare *value equality* in Java, you probably want to use `object.equals()` instead of using the `==` operator.

In the Gosu language, the `==` operator automatically calls `object.equals()` for comparison if you use it with reference types. In most cases, this is what you want for reference types.

However, there are some cases in which you want to use identity reference, not simply comparing the values using the underlying `object.equals()` comparison. In other words, some times you want to know if two objects literally reference the same in-memory object.

In ClaimCenter 6.0, you can now use the new object equality operator `===` (three equals signs). This always compares whether both references point to the same in-memory object.

The following example demonstrates the difference between `==` and `===` operators:

```
var s1 = "Hello"
var s2 = "Hello"

// compare value
print( s1 == s2 ) // prints true -- the two String objects contain the same contents

// compare identity
print( s1 === s2 ) // prints false -- the two String objects are DIFFERENT objects

// assign the s1 variable to the EXACT same object as in s2
s1 = s2

// compare identity
print( s1 === s2 ) // prints true -- these two variables now reference the same object!
```

Object Initializer Syntax During Object Creation

Object initializers allow you to set properties on an object immediately after a new expression. In other words, you can assign properties as part of creating a new object. Use object initializers for compact and clear object declarations. They are especially useful if you combine them with data structure syntax and nested objects.

A simple version looks like the following:

```
var sampleClaim = new Claim(){ :ClaimId = "TestID" }
```

The syntax is as follows. After a constructor, open a curly brace. Next, add a clause containing a colon, followed by a property name, an equals sign, and the property value. Repeat this clause for any additional properties, delimiting multiple clauses with commas. In other words, each name/value pair has the following syntax:

```
:PROPERTY_NAME = VALUE
```

Remember to add a colon before the property name.

For example, suppose you have the following code:

```
var myFileContainer = new my.company.FileContainer()
myFileContainer.DestFile = jarFile
myFileContainer.BaseDir = dir
myFileContainer.Update = true
myFileContainer.WhenManifestOnly = ScriptEnvironment.WHEN_EMPTY_SKIP
```

You can use object initializers to simplify this code to the following:

```
var myFileContainer = new my.company.FileContainer() { :DestFile = jarFile, :BaseDir = dir,
:Update = true, :WhenManifestOnly = ScriptEnvironment.WHEN_EMPTY_SKIP }
```

This syntax also is useful for naturally expressing nested object trees, such as XML data.

For example, suppose you have the following code:

```
using xsd.test.*

var simpleTest = new SimpleTest()
simpleTest.id = "Root"

var test2 = new Test2()
test2.id = "test"

simpleTest.test2s.add(test2)
simpleTest.test2s.add(new Test2())
simpleTest.test2s.get(1).final = true
simpleTest.test2s.get(1).Test1 = new TestType()

var test1 = new xsd.test.TestType()
test1.color = Red; // Note that Gosu can infer what enum class is appropriate!
test1.number = 5
```

```

simpleTest.test4s.add(test1)
simpleTest.test3 = Blue // Since this is a simple child element, you access its value directly

return simpleTest.toXML()

```

You can naturally express it compactly as the following Gosu code:

```

using xsd.test.*

var simpleTest = new SimpleTest(){ :id = "Root", :test3 = Blue,
                                   :test2s = { new Test2(){ :id = "test" },
                                                new Test2(){ :final = true, :Test1 = new TestType() } },
                                   :test4s = { new TestType(){ :color = Red, :number = 5 } }
                                   }

print( simpleTest.toXML())

```

The object initializer syntax more clearly expresses the nested nature of XML nodes. Each node clearly defines what the generated XML would look like.

For more information, see “Object Initializer Syntax” on page 71 in the *Gosu Reference Guide*.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Review your code to use the new compact object initializer syntax. This makes your Gosu code easier to read and maintain.

Type System Reflection New APIs

There are new supported APIs for advanced introspection into types at run time, a feature called *reflection*. For more information about these APIs in the `gw.lang.reflect.*` package, refer to “Type System” on page 299.

New Stream Utilities

There are new utilities for writing Gosu code that uses data streams. Refer to the Help in Studio for more information about the class `gw.util.StreamUtil`.

Concurrency Utilities

There are new classes for supporting concurrency in Gosu. These are valuable for supporting multi-threaded code successfully with shared data across threads. This is particularly important for code that relies on static class variables. For more information about the new APIs in the package `gw.util.concurrent.*`, see “Concurrency” on page 311.

Checksum APIs

There are new classes for generating checksums (fingerprints) on objects from Gosu. Use checksum APIs to improve detection from accidental modification of data in transit. For more information, see “Checksums” on page 329.

Run With New Bundle With a User

The `gw.transaction.Transaction` class has a new alternate version of the `runWithNewBundle` method to create a bundle with a specific user associated with it. Use this in contexts in which there is no inherent user context, such as in *startable plugins*. The method signature is:

```
gw.transaction.Transaction.runWithNewBundle(\ bundle -> YOUR_BLOCK_BODY, user)
```

For that second method argument, you can pass either a `User` entity or a `String` that is the user name.

Changed in Gosu in 6.0

This topic describes changes in Gosu for ClaimCenter 6.0. These include:

- “Renamed GScript to Gosu” on page 60
- “Java Collection Generics Fixes” on page 60
- “Data Structure Declaration Extensions for Lists, Maps, and Sets” on page 61
- “Class Variables Can Both Expose as Property and Have Initializer” on page 61
- “Compound Types” on page 62
- “Moved Enhancement Packages” on page 63
- “XML Node Package Name Changed” on page 63
- “Gosu Error Reporting Improvements” on page 63
- “Improvements to XSD and XML Processing” on page 63
- “Changes to XSD Class Loading Behavior” on page 64
- “Query Builder API Changes” on page 65
- “Query Builder API Improvements” on page 65
- “Blocks Require Single Statements in Braces” on page 66
- “Changes to Existing Collections Enhancement Methods” on page 67
- “New Collections Enhancement Methods” on page 68
- “New Array/List Expansion Operator (Deprecated Old Style)” on page 71
- “Gosu Array Enhancement Changes” on page 72
- “New Concurrency and Scoping APIs, Scoped Variables Deprecated” on page 73
- “New ‘Type’ Property on All Types” on page 73
- “Exception Changes If No Current User and Creating New Bundle” on page 74
- “Block Declarations Now Require Argument Names” on page 74
- “Function Pointers and Nested Functions Now Unsupported” on page 74

Renamed GScript to Gosu

As of ClaimCenter 6.0, the GScript language is called Gosu. This affects not only the documentation, but all APIs, file names, or package names. Any use of the word GScript becomes Gosu. Any use of the lowercase word gscript becomes gosu.

Moved Classes Hierarchy

In ClaimCenter 5.0, GScript classes were in the following location:

`ClaimCenter/modules/configuration/config/resources/classes/...`

In ClaimCenter 6.0, Gosu classes (formerly called GScript classes) are in the following location:

`ClaimCenter/modules/configuration/gsrc/...`

Java Collection Generics Fixes

Some methods on Java collection classes in Gosu now take the correct parameter types. Specifically, they take strongly typed parameter using generics instead of taking an argument of type `Object`.

For example, the `List<String>` method `contains(o : Object)` method now has the signature `contains(s : String)`, which is more typesafe. This works because Gosu does not use the Java notion of wildcards, instead using array-like covariance for generic types.

This affects the following:

```
Collection.contains(E)
Collection.remove(E)
Collection.removeAll(E)
Collection.containsAll(E)
Collection.removeAll(E)
List.indexOf(E)
List.lastIndexOf(E)
Map.get(K)
Map.containsKey(K)
Map.remove(K)
Map.containsValue(V)
```

For more information about collections methods, see “Collections” on page 231 in the *Gosu Reference Guide*.

Data Structure Declaration Extensions for Lists, Maps, and Sets

In earlier releases of Gosu, you could write the following code to initialize list values:

```
var x : List<String> = {"a", "b", "c"}
```

Note: The angle bracket notation indicates support for parameterized types, using Gosu generics features. For more information, refer to “Gosu Generics” on page 221.

You could also use this syntax:

```
var x = new List<String>() {"a", "b", "c"}
```

However, you could not write the following:

```
var x = {"a", "b", "c"}
```

Gosu now permits this last example as valid and is typesafe. Gosu infers the type of the `List` to be the least upper bound of the components of the list. In the simple case above, the type of the variable `x` at compile time is `List<String>`. If you pass different types of objects, Gosu finds the most specific type that includes all of the items in the list.

If the types implement interfaces, Gosu attempts to preserve commonality of interface support in the list type. This ensures your list acts as expected with APIs that rely on support for the interface. In some cases, the resulting type is a *compound type*, which combines a *class* and one or more *interfaces* into a single type. For example, the following code initializes an `int` and a `double`:

```
var s = {0, 3.4}
```

The resulting type of `s` is `ArrayList<java.lang.Comparable & java.lang.Number>`. This means that it is an array list of the compound type of the class `Number` and the interface `Comparable`.

Note: The `Number` class does not implement the interface `Comparable`. If it did, then the type of `s` would simply be `ArrayList<java.lang.Number>`. However, since it does not implement that interface, but both `int` and `double` implement that interface, Gosu assigns the compound type that includes the interfaces that they have in common.

This new type inference works with maps, as shown in the following examples:

```
var numbers = {0 -> 1, 3 -> 3.4}
var strings = {"a" -> "value"}
```

The type inference also works with sets, as shown in the following example:

```
var s : Set = {1,2,3}
```

Also see related section “Compound Types” on page 62.

Class Variables Can Both Expose as Property and Have Initializer

You can use a new simpler syntax to expose a class level variable as a property using the `as` keyword and also have an initializer expression:

```
class MyClass {  
    var _myPrivateName : String as Name = "Default"  
}
```

In previous releases, these two features were mutually exclusive.

Compound Types

Gosu now supports *compound types* (sometimes known in other languages as intersection types). Compound types are combinations of two distinct types to create a new type. This is a fairly specialized need, but is required in some special cases. A compound type is the result of taking two or more types and determining their least upper bound type of two disjoint types that share multiple common interfaces.

In some cases, the resulting type is a *compound type*, which combines a *class* and one or more *interfaces* into a single type. Gosu supports compound types to support other Gosu features, such as the *delegate* keyword and a special concise syntax for generating lists, sets, and maps of varied types.

If Gosu combines two types that implement interfaces, Gosu attempts to preserve the commonality of interface support. This means the resulting object acts as expected with APIs that rely on support for the interface. In some cases, the resulting type is a *compound type*, which combines a *class* and one or more *interfaces* into a single type. For example, the following code initializes an `int` and a `double`:

```
var s = {0, 3.4}
```

The resulting type of `s` is `ArrayList<java.lang.Comparable & java.lang.Number>`. This means that it is an array list of the compound type of the class `Number` and the interface `Comparable`.

For more information about compound types, see “Compound Types” on page 308 in the *Gosu Reference Guide*.

Change in Ternary Conditionals If Clauses Return Different Types

A ternary conditional checks a condition and returns one of two values depending on whether the expression returns true or false. The behavior changed in a subtle way if the true and false clauses return different values. The compile time type of the result may change.

In previous releases, the compile-time type of the ternary conditional was always the value of the first clause, the true clause.

In this release, if the true clause and the false clause return different types, the result is a combination of the types of both clauses. For example, consider the following statement:

```
var s = someCondition ? "hello" : false
```

The type of the result is the type lowest down in the type hierarchy to contain both types. If either clause is a primitive type such as `int` or `boolean`, Gosu coerces the primitive type to its boxed (subclass of `Object`) version before doing this change. For example, `boolean` coerces to `Boolean`.

If they have no ancestors in common, the compile-time type of the result is `Object`. This is important to note, because it may affect the coercions you do with the result and what properties or methods you call on the result.

Note: Although the compile-time type looks at both types to find a common ancestor in the type hierarchy, a ternary clause does not cause the creation of a Gosu *compound type*. For a more detailed explanation of this special type, see “Compound Types” on page 308

If the true or false clauses return the same type, nothing changes in this release.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Review your code for ternary condition expressions (`condition ? trueValue : falseValue`). Check for any cases in which you pass different types to the true clause and the false clause. If you did not declare the variable type, check for possible changes to the compile-time type of the result.

As part of upgrade, you must carefully review your use of ternary expressions in cases where the types differ between the `true` clause and the `false` clause.

WARNING Review your code for ternary condition expressions in which you pass different types to the `true` clause and to the `false` clause and do not declare the variable type. If you did not declare the variable type, check for possible changes to the result in the `false` clause coercion or implications of the change to the variable's type.

For a related change, see “Data Structure Declaration Extensions for Lists, Maps, and Sets” on page 61.

Moved Enhancement Packages

Some core Gosu enhancements of the Gosu programming language (such as `CoreListEnhancement.gsx`) moved to a different package: `gw.lang.enhancements.*`. Generally speaking, this does not affect you unless you modified the built-in enhancements. Enhancements are extra methods and properties that you can add from Gosu. Enhancements work with types even if you do not have access to the original type source, or if the type is a Java type.

UPGRADE TASK

Priority: Before starting the server

Summary: If you modified built-in Gosu enhancements to lists, update your code to merge changes into the new package location

XML Node Package Name Changed

In previous releases, Gosu included native support for XML objects files based on the class `gw.api.xml.XMLNode`. This changed packages to the new fully-qualified name `gw.xml.XMLNode`.

All existing XML parsing code that uses the old package names continue to work. This is possible because of internal changes to the Gosu type system that aliases the old type names to point to the new types in their new locations.

For any new code, use the new name `gw.xml.XMLNode` instead of `gw.api.xml.XMLNode`. Guidewire recommends updating code to the new names, but it is not an upgrade requirement in this release.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Change any references to `gw.api.xml.XMLNode` to instead use `gw.xml.XMLNode`.

Gosu Error Reporting Improvements

Gosu compiler errors improved in this release. Errors are more likely to describe the true problem.

Improvements to XSD and XML Processing

This release contains improvements to Gosu XSD and XML processing. For more information, see “Gosu and XML” on page 245 in the *Gosu Reference Guide*.

Namespace Improvements

The XML parser supports XSD namespaces using QNames. A QName stands for *qualified name*. The XSD standard uses QNames to represent objects within a specific namespace. A QName property on `XMLNode` objects sets or gets the namespace information. Gosu handles namespace conflicts automatically at serialization time. If you

set an `xsd:QName` property, Gosu automatically declares the namespace at the appropriate level during serialization if needed. In the case of prefix conflicts, Gosu uses a naming scheme of `ns0`, `ns1`, `ns2`, and so on. For more information, see “XSD Namespaces and QNames” on page 260 in the *Gosu Reference Guide*.

`xsd:IDREF`, `xsd:IDREFS`, and `xsd:ID`

If a schema contains an element or attribute of type `xsd:IDREF`, it becomes a property of type `IXMLNodeWithID`. Any element with an attribute of type `xsd:ID` will automatically implement `IXMLNodeWithID`. By assigning an `IXMLNodeWithID` to an `IDREF` property, the two elements become linked on output with no danger of infinite recursion. IDs are automatically assigned if not set explicitly, and a document that is parsed then reserialized will retain the original ID values unless there are ID conflicts within the document. Be aware that ID references are not typed. This means you must cast the `IXMLNodeWithID` to the appropriate `XMLNode` type before you can take further action on it other than usual `IXMLNode` actions. For more information, see “XML Node IDs” on page 260 in the *Gosu Reference Guide*.

Autocreation of XSD-based XMLNode properties

Properties on XSD-Based XML nodes referring to other XSD-Based XML nodes automatically create if they do not exist. This allows you to assign to very long paths (dot-paths) without manually creating intermediate nodes. For more information, see “Autocreation of Intermediate Nodes” on page 260 in the *Gosu Reference Guide*.

`xsd:dateTime` and related types

All of the XSD types related to dates (`xsd:dateTime` and related types) have Gosu counterparts in the package `gw.api.xml.xsd.types`. All of them share the common superclass `AbstractXSDDateType`. You can construct all of them from a `java.lang.String` in the standard XML Schema date/time formats specified in the W3C XML Schema specification. The `toString` method on any of them return the same format. You can alternatively construct one of these from a `java.util.Calendar` object, passing a `boolean` specifying whether or not to include the `Calendar` object's time zone. The meaning of an `xsd:dateTime` without time zone information is application-specific. You can call the `toCalendar` method on a date type to convert back to a calendar object. However, Gosu throws an `IllegalStateException` if you do not include time zone information. You can call the `date.toCalendar(tz : TimeZone)` method to specify a default time zone to use if a time zone does not exist in the `xsd:dateTime` object. Gosu sets all `Calendar` fields to defaults except the fields that are significant to the `xsd:dateTime`. For more information, see “XSD Namespaces and QNames” on page 260 in the *Gosu Reference Guide*.

Posting an XML Graph to a URL

This convenience method posts the receiver to a specified URL and returns the response as a byte array (`byte[]`). You can pass the byte array passed to an `XMLNode.parse()` method to generate another `XMLNode` graph based on that. The method signature looks like the following:

```
IXMLNode.postToURL(url : String) : byte[]
```

While not full-featured enough for a production environment, you can use this method for quick prototypes, demonstrations, and experimentation.

Changes to XSD Class Loading Behavior

In ClaimCenter 5.0, you put your XSDs in the following location:

```
ClaimCenter/modules/configuration/config/registry/xsds
```

Next, you would register your new XSD in the Gosu XSD registry file at the location:

```
ClaimCenter/modules/configuration/config/registry/registry-xsd.xml
```

In ClaimCenter 6.0, the XSD registration changed. You can now simply put your XSD files in the same file hierarchy in the configuration module as you create Gosu classes. The **location** of your XSD in the class hierarchy defines the package into which Gosu loads your XSD. For example, suppose you add an XSD file at the path:

```
ClaimCenter/modules/configuration/gsrc/mycompany/messaging/MyData.xsd
```


When Studio starts, Gosu loads objects in this XSD in the namespace:

```
mycompany.messaging.MyData.*
```

If you use the old-style XSD registration system (including `registry-xsd.xml`), Gosu loads the types in the XSD as in previous releases. However, all those types appear in Gosu as deprecated types in Studio.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: If you use the old-style XSD registration system (including `registry-xsd.xml`), update your code to use the new XSD class loading system. This avoids your Gosu code using now-deprecated types.

XSDs that Reference Other XSDs Through HTTP

For maximum performance and reliability, the Gosu XSD type loader does not follow HTTP links. If an XSD references non-local resources such as XSDs accessed through HTTP, you must copy those resources to a local directory. For example, if an XSD references a remote XSD, place a copy of the remote XSD into the same directory as your schema in the Gosu class hierarchy. Otherwise, Gosu cannot parse your main XSD. You do **not** need to modify either XSD to support locally loading the XSD. Instead, Gosu looks for a local resource with the same name as that XSD in the same directory instead of following the HTTP link.

Query Builder API Changes

In ClaimCenter 6.0, the existing query builder API changed in ways that affect existing code during upgrades.

IMPORTANT The query builder is the new database query system introduced in ClaimCenter 5.0.5 that replaces `find` expressions. You can still use `find` expressions. The `find` expression syntax is not deprecated. However, Guidewire **strongly recommends** that for any new code that you write, use the new query builder APIs. For more information about query builder, see “Query Builder” on page 129 in the *Gosu Reference Guide*.

The type of a query processor changed, as well as the way to get a query processor.

In previous releases:

- call `query.getProcessor()` to get an instance of `QueryProcessor`

In ClaimCenter 6.0:

- call `query.select()` to get an instance of `IQueryResult`

For related changes, see “Query Builder API Improvements” on page 65.

Query Builder API Improvements

In ClaimCenter 6.0, the query builder APIs improved compared to ClaimCenter 5.0.x.

IMPORTANT The query builder is the new database query system introduced in ClaimCenter 5.0.5 that replaces `find` expressions. You can still use `find` expressions. The `find` expression syntax is not deprecated. However, Guidewire **strongly recommends** that for any new code that you write, use the new query builder APIs. For more information about query builder, see “Query Builder” on page 129 in the *Gosu Reference Guide*.

In many situations you only need a few columns from the result instead of the whole entity. You can use a variant of the `select` method on a query, which replaces the `query.getProcessor()` method. See “Query Builder API Changes” on page 65 for details.

Just as using `getProcessor` in ClaimCenter 5.0, you can call the method with no arguments to simply use it as an iterator across all results. Each query result is the entity in its entirety, and the database loads all columns in that row.

Instead of calling the `select` method with no arguments, in ClaimCenter 6.0, you can now use an alternate method signature that takes a *block*. A block is an in-line function that you define within another function. For more information about blocks, see “Gosu Blocks” on page 213 in the *Gosu Reference Guide*. You pass a block that takes an object of the query type and returns whatever you want. The return type can be anything that you specify. Any arbitrary type is fine, so you can choose whatever is most convenient for your program. The return type does not need to match the native types for the columns in the database.

This alternative `select` method provides the following important benefits:

- **Gosu only loads the columns you need.** The database returns **only** the columns you access in your block. This reduces memory usage and improves overall server performance.
- **You can assemble arbitrary structures or calculations to return in place of full entities.** Since you are not retrieving the entire entity, you choose how to return each result. For example, you can populate a custom Gosu class containing only the fields you need. Or you can simply return a single field that you need, such as a public ID string.

For more information, see “Selecting Columns and Returning Results in Custom Formats” on page 145.

Additionally, ClaimCenter improved the ability to order results by database fields, including multi-level field paths. You can now select second-level or additional levels of sorting. You can choose the order (forward or reverse) for each level of sorting. For example, the following orders by column A2 then by descending order on column E.E:

```
result.orderBy(\row-> row.A2).thenByDescending(\row->row.E.E)
```

For more information, see “Ordering results” on page 147

Blocks Require Single Statements in Braces

Gosu blocks no longer accept single statements as bodies. You must wrap single-statement in curly braces to turn it into a statement list. This change does not affect the ability to pass an *expression* without using braces. However, this might affect some rare cases in which you use an assignment statement, for instance.

For example, in ClaimCenter 5.0, the following was a valid block:

```
\ n -> capturedVariable = n // An assignment statement, not an expression body!
```

In ClaimCenter 6.0, this must become the following:

```
\ n -> { capturedVariable = n } // You must wrap the statement in { and } to become a statement list
```

The compound assignment operations that are new in ClaimCenter 6.0 are always statements not expressions. Thus, the following expression is invalid:

```
names.each( \ n -> myValue += n )
```

Instead, change it to the following:

```
names.each( \ n -> { myValue += n } )
```

This also affects other statements such as `if` and `while` statements. For example, in ClaimCenter 5.0, the following was a valid block:

```
\ n -> if ( n < 100 ) {print("low")} else {print("high")}
```

In ClaimCenter 6.0, this must become the following:

```
\ n -> {if ( n < 100 ) {print("low")} else {print("high")}}
```

This change does **not** affect the ability to pass an expression. You can still freely pass an expression as a block body without using braces:

```
\ n -> n + 1
```

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Check for compile errors due to the new requirement that Gosu blocks cannot contain single statements. To fix this, add braces around the statement. This does not affect expressions as the body of a Gosu block.

Changes to Existing Collections Enhancement Methods

The Gosu APIs related to collections changed. ClaimCenter implements these APIs using Gosu enhancements that you can view in Studio. The following changes affect existing collection code. The old names for these methods still work in this release but are deprecated.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Change all uses of deprecated collection methods to the renamed versions. Affects: `countMatches`, `findFirst`, `findAll`, `findByType`, `removeMatches`, `keepMatches`. The replacement for `findFirst` changes behavior with empty sets. While renaming methods, carefully review all code that uses `findFirst` to ensure you do not rely on the old behavior.

'countMatches' becomes 'countWhere'

The collection method `countMatches(cond)` method is now `countWhere(cond)`.

Suppose you have code like this:

```
var totalWithOtherCoverage = claim.Exposures.countMatches( \ exp -> exp.OtherCoverage )
```

Change it to:

```
var totalWithOtherCoverage = claim.Exposures.countWhere( \ exp -> exp.OtherCoverage )
```

'findFirst' becomes 'firstWhere' with New Empty Set Behavior

The collection method `findFirst(cond)` method is now `firstWhere(cond)`.

Suppose you have code like this:

```
var firstExpWithOtherCoverage = claim.Exposures.findFirst( \ exp -> exp.OtherCoverage )
```

Change it to:

```
var firstExpWithOtherCoverage = claim.Exposures.firstWhere( \ exp -> exp.OtherCoverage )
```

There is a difference in the behavior in the new method.

The `firstWhere` method now throws an exception if no objects match the given condition.

'findAll' becomes 'where'

The collection method `findAll(cond)` method is now `where(cond)`.

Suppose you have code like this:

```
var allExpWithOtherCoverage = claim.Exposures.findAll( \ exp -> exp.OtherCoverage )
```

Change it to:

```
var allExpWithOtherCoverage = claim.Exposures.where( \ exp -> exp.OtherCoverage )
```

'findByType' becomes 'whereTypeIs'

The collection method `findByType(ITYpe)` method is now `whereTypeIs(ITYpe)`.

Suppose you have code like this:

```
var wcExposures = claim.Exposures.findByType( WokersCompExposure )
```

Change it to:

```
var wcExposures = claim.Exposures.whereTypeIs( WokersCompExposure )
```

'removeMatches' becomes 'removeWhere'

The collection method `removeMatches(cond)` method is now `removeWhere(cond)`.

This is a rename and also a return type change. This method now returns no value, so it cannot be chained like the older method. This changes makes clear that a mutation happens **in place** rather than creating a new Collection with all offending elements removed.

Suppose you have code like this:

```
var inactiveCoverages = coverages.removeMatches( \ cov -> not cov.Active )
```

Change it to:

```
var inactiveCoverages = coverages.copy()
// IMPORTANT: the mutation occurs in place
inactiveCoverages.removeWhere( \ cov -> not cov.Active )
```

Ideally, the following is an even better version of this code using the `where` method:

```
var inactiveCoverages = coverages.where( \ cov -> not cov.Active )
```

'keepMatches' becomes 'retainWhere'

The collection method `keepMatches(cond)` method is now `retainWhere(cond)`.

This is a rename and also a return type change. This method now returns no value, so it cannot be chained like the older method. This is to make clear that the mutation is happening in place, rather than a new Collection being created with the offending elements removed.

Suppose you have code like this:

```
var activeCoverages = coverages.keepMatches( \ cov -> cov.Active )
```

Change it to:

```
var activeCoverages = coverages.copy()
activeCoverages.retainWhere( \ cov -> cov.Active ) // the mutation occurs in place
```

Ideally, the following is an even better version of this code:

```
var activeCoverages = coverages.where( \ cov -> cov.Active )
```

New Collections Enhancement Methods

Gosu contains new enhancement methods for collections and related types.

Methods on Iterable<T>

Iterable objects (objects that implement `Iterable<T>`) have additional methods described in the following table. The following table lists the available methods, including methods that are unchanged so you can easily understand the changes. The second column indicates whether this method is new, unchanged from previous releases, or renamed compared to ClaimCenter 5.0.

Method/Property	New, Unchanged, or Renamed	Old method name, if applicable	Description
<code>Count</code>	New		Returns the number of elements in the <code>Iterable</code>
<code>single()</code>	New		If there is only one element in the <code>Iterable</code> , that value is returned. Otherwise an <code>IllegalStateException</code> is thrown.

Method/Property	New, Unchanged, or Renamed	Old method name, if applicable	Description
<code>toCollection()</code>	New		If this Iterable is already of type <code>Collection</code> , return it. Otherwise, copy all values out of this Iterable into a new <code>Collection</code> .

Methods on `Collection<T>`

Most collection methods are now implemented directly on `Collection`, not `List` or other similar objects as in previous releases. The following table lists the available methods, including unchanged methods so you can better understand the changes. The second column indicates whether this method is new, unchanged from previous releases, or renamed compared to ClaimCenter 5.0.

Method/Property Name	New, Unchanged, or Renamed	Old method name, if applicable	Description
<code>allMatch(cond)</code>	New		Returns true if all elements in the <code>Collection</code> satisfy the condition
<code>hasMatch(cond)</code>	Unchanged		Returns true if this <code>Collection</code> has any elements in it that match the given block
<code>asIterable()</code>	New		Returns this <code>Collection<T></code> as a pure <code>Iterable<T></code> (in other words, not as a <code>List<T></code>).
<code>average(selector)</code>	New		Returns the average of the numeric values selected from the <code>Collection<T></code>
<code>countWhere(cond)</code>	Renamed	<code>countMatches()</code>	Returns the number of elements in the <code>Collection</code> that match the given condition
<code>HasElements</code>	New		Returns true if this <code>Collection</code> has any elements in it. This is a better method to use than the default collection method <code>empty()</code> because <code>HasElements</code> interacts better with null values. For example, the expression <code>col.HasElements()</code> returns a non-true value even if the expression <code>col</code> is null.
<code>first()</code>	New		Returns first element in the <code>Collection</code> , or return null if the collection is empty.
<code>firstWhere(cond)</code>	Renamed	<code>findFirst(cond)</code>	Returns first element in the <code>Collection</code> that satisfies the condition, or throws an exception if none do.
<code>flatMap(proj)</code>	Unchanged		Maps each element of the <code>Collection</code> to a <code>Collection</code> of values and then flattens them into a single <code>List</code> .
<code>fold()</code>	Unchanged		Accumulates the values of an <code>Collection<T></code> into a single <code>T</code> .
<code>intersect(iter)</code>	Unchanged		Returns a <code>Set<T></code> that is the intersection of the two <code>Collections</code>
<code>last()</code>	Unchanged		Returns last element in the <code>Collection</code> or return null if the list is empty
<code>lastWhere(cond)</code>	New		Returns last element in the <code>Collection</code> that matches the given condition, or null if no elements match it
<code>map(proj)</code>	Unchanged		Returns a <code>List</code> of each element of the <code>Collection<T></code> mapped to a new value.
<code>max(proj)</code>	New		Returns maximum of the selected values from <code>Collection<T></code>
<code>min(proj)</code>	New		Returns minimum of the selected values from <code>Collection<T></code>

Method/Property Name	New, Unchanged, or Renamed	Old method name, if applicable	Description
<code>orderBy(proj)</code>	New		Returns a new <code>List<T></code> ordered by the given value. Note that this is different than <code>sortBy()</code> , which is retained on <code>List<T></code> and which sorts in place.
<code>orderByDescending(proj)</code>	New		Returns a new <code>List<T></code> reverse ordered by the given value. Note that this is different than <code>sortByDescending()</code> , which is retained on <code>List<T></code> and which sorts in place.
<code>partition(proj)</code>	Unchanged		Partitions this <code>Collection</code> into a Map of keys to a list of elements in this <code>Collection</code> .
<code>partitionUniquely(proj)</code>	Unchanged		Partitions this <code>Collection</code> into a Map of keys to elements in this <code>Collection</code> . Throws an <code>IllegalStateException</code> if more than one element maps to the same key.
<code>reduce(init, reducer)</code>	Unchanged		Accumulates the values of a <code>Collection<T></code> into a single value given an initial seed value.
<code>reverse()</code>	Unchanged		Reverses the collection as a <code>List</code> .
<code>singleWhere(cond)</code>	New		If there is only one element in the <code>Collection</code> that matches the given condition, it is returned. Otherwise an <code>IllegalStateException</code> is thrown
<code>sum(proj)</code>	New		Returns the sum of the numeric values selected from the <code>Collection<T></code>
<code>thenBy(proj)</code>	New		Additionally orders a <code>List</code> that has already been ordered by <code>orderBy</code> .
<code>thenByDescending(proj)</code>	New		Additionally reverse orders a <code>List</code> that has already been ordered by <code>orderBy</code> .
<code>toList()</code>	Unchanged		If this <code>Collection</code> is already a list, simply return it. Otherwise create a new <code>List</code> and copy this <code>Collection</code> to it.
<code>toTypedArray()</code>	Unchanged		Converts this <code>Collection<T></code> into an array <code>T[]</code> .
<code>union(col)</code>	Unchanged		Returns a new <code>Set<T></code> that is the union of the two <code>Collections</code>
<code>where(cond)</code>	Renamed	<code>findAll()</code>	Returns all elements in this <code>Iterable</code> that satisfy the given condition
<code>whereTypeIs(Type)</code>	Renamed	<code>findByType()</code>	Returns a new <code>List<T></code> of all elements that are of the given type
<code>disjunction()</code>	New		Returns a new <code>Set<T></code> that is the set disjunction of this collection and the other collection
<code>each()</code>	New		iterates each element of the <code>Collection</code>
<code>eachWithIndex()</code>	New		Iterates each element of the <code>Collection</code> with an index
<code>join</code>	New		joins all elements together as a string with a delimiter
<code>minBy()</code>	New		Returns the minimum <code>T</code> of the <code>Collection</code> based on the projection to a <code>Comparable</code> object
<code>maxBy()</code>	New		Returns the maximum <code>T</code> of the <code>Collection</code> based on the projection to a <code>Comparable</code> object
<code>removeWhere()</code>	Renamed	<code>removeMatches</code>	Removes all elements that satisfy the given criteria
<code>retainWhere()</code>	New	<code>keepMatches()</code>	Removes all elements that do not satisfy the given criteria. This method returns no value, so it cannot be chained in series. This is to make clear that the mutation is happening in place, rather than a new collection created with offending elements removed.
<code>subtract()</code>	New		Returns a new <code>Set<T></code> that is the set subtraction of the other collection from this collection

Method/Property Name	New, Unchanged, or Renamed	Old method name, if applicable	Description
toSet()	New		Converts the Collection to a Set

Methods on List<T>

Method/Property Name	New, Unchanged, or Renamed	Old Method Name	Description
reverse()	New		Reverses the Iterable.
copy()	New		Creates a copy of the list
freeze()	New		Returns a new unmodifiable version of the list
shuffle()	New		Shuffles the list in place
sort()	New		Sorts the list in place
sortBy()	New		Sorts the list in place in ascending order
sortByDescending()	New		Sorts the list in place in descending order

New Methods on Set<T>

Method/Property Name	New, Unchanged, or Renamed	Old Method Name	Description
copy()	New		Creates a copy of the set
powerSet()	New		Returns the power set of the set
freeze()	New		Returns a new unmodifiable version of the set

New Array/List Expansion Operator (Deprecated Old Style)

Gosu now includes a new operator for array expansion and list expansion. It is an asterisk followed by a period, for example:

```
names*.Length
```

In ClaimCenter 5.0, Gosu enabled array expansion by overloading the member access operator (the dot operator). For example, consider the `User` entity. Suppose you wanted to get a list of the groups a user belongs to so you can display the display names of each group. The `User` entity contains a `MemberGroups` property that returns a read-only array of groups that the user belongs to. In other words, the Gosu syntax `user.MemberGroups` returns an array of `Group` entities. If you want to get the display names from each group, you can use the following Gosu code

```
var user : entity.User
...
print( user.MemberGroups.DisplayName )
```

Because `MemberGroups` is an array, Gosu implicitly expands the array by the `DisplayName` property on the `Group` component type. The result is an array of the names of all the Groups to which the user belongs. The type is `String[]`. This array and list expansion can be useful and powerful.

However, there are drawbacks to this approach:

- The subtle syntax results in programmers unaware of this feature.
- Because it uses the standard dot notation, even programmers familiar with the feature might accidentally use the feature when unintended. Also, programmers might not instantly understand the complexity of the expansion implicit in the code. For example, if you do not notice that `MemberGroups` is plural and you let the editor complete the expression, you might expect the result to be type `String`, not `String[]`.

- Properties on the array may conflict with properties on the array component. For instance, the `Length` property on array conflicts with the `Length` property on `java.util.String` (a Gosu enhancement adds the `Length` property). Thus, the following expression is ambiguous:

```
var names : String[]
...
print( names.Length )
```

Gosu cannot know whether the programmer intended to return an integer representing the length of the array or an array containing lengths of the names. Currently, array properties have precedence over component type properties, so Gosu return an integer representing the length of the array. However, this also means there is no concise way to return the array containing lengths of the names with array expansion.

In ClaimCenter 6.0, Gosu solves most of these issues by introducing a new operator exclusive to array expansion: the new `*` expansion operator. For example, the expressions from above can be expressed as follows:

```
// An array of names for each group
var arrayOfGroupNames = user.MemberGroups*.DisplayName

// An array of lengths for each name
var lengthOfNames = names*.length
```

The expansion operator works with methods also:

- If the object to the left of the expansion operator is an array, Gosu gets the type that the method returns and creates an array of that type.
- If the object to the left of the expansion operator is a list, Gosu gets the type `T` that the method returns and creates a list of that type. Using Gosu generics notation, the result type is `java.util.List<T>`. The following example calls a method on the `String` component of the `List` of `String` objects. It generates the list of initials.

```
var s = {"Fred", "Garvin"}

// get the character array [F, G]
var charArray = s*.charAt( 0 )
```

Other important notes about the new expansion operator:

- Older code is backward-compatible.
- The older style is now deprecated. You can use it in this release, but it is best to start to convert your code to the new expansion operator.
- The new expansion operator `*` works only for array expansion, never standard property accessing.
- When using the new expansion operator, only **component type properties** are accessible.
- When using the new expansion operator, **array properties** are never accessible.
- When using the legacy syntax for expansion (the simple period), component type properties do not pollute the array type information within in code completion in the Gosu editor.
- When using the legacy syntax for expansion (the simple period), the compiler warns you about ambiguity.
- The expansion operator applies not only to arrays, but to any `Iterable` type and all `Iterator` types and it preserves the type of array/list. For instance, if you apply the `*` operator to a `List`, the result is a `List`. Otherwise, the expansion behavior is the same as with arrays.

For more information, see “List and Array Expansion (`*`)” on page 235 in the *Gosu Reference Guide*.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Review your code for uses of the now-deprecated list/array expansion operator (the single period). Replace these with the new `*` operator syntax.

Gosu Array Enhancement Changes

Gosu enhancements on arrays no longer use the special syntax `gw.lang.parser.DummyArrayType<T>`.

You can now use the more straightforward syntax: `T[]`

The old `DummyArrayType<T>` style is now unsupported.

This change does not affect code that uses the enhancements, only the definition of the enhancement itself.

For more information, see “Enhancements on Arrays” on page 212 in the *Gosu Reference Guide*.

UPGRADE TASK

Priority: Before starting the server

Summary: If you wrote any Gosu enhancements on arrays, update the enhancement definition file with the updated syntax.

New Concurrency and Scoping APIs, Scoped Variables Deprecated

As of ClaimCenter 6.0, the direct variable scoping syntax is deprecated. As a result, **all** variable scoping modifiers are deprecated. This includes the following Gosu variable scoping keywords: `execution`, `request`, `session`, and `application`.

Variable scoping is most useful for static variables. However, Guidewire strongly recommends you do not rely on variable scoping.

There are several systems you can use to replace this functionality:

- New scope-related utilities in the new class `gw.api.web.Scopes`. These help synchronize and protect access to shared data using synchronized Map objects you can get and put your variables for different scopes.
- The `LazyVar` class (in `gw.util.concurrent`) 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. The `LazyVar` class wraps the double-checked locking pattern in a typesafe holder. For more information, see “Concurrent Lazy Variables” on page 314 in the *Gosu Reference Guide*.
- The `Cache` class (in `gw.util.concurrent`) declares a cache of values you can look up quickly and in a thread-safe way. It declares a concurrent cache similar to a Least Recently Used (LRU) cache. Because the `Cache` class uses the Java concurrency libraries, access to the concurrent cache is thread-safe. For more information, see “Concurrent Cache” on page 314 in the *Gosu Reference Guide*.
- The standard Java concurrency libraries.

For more details, see “Concurrency” on page 311 in the *Gosu Reference Guide*.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Remove any deprecated variable scoping modifiers and replace with new scoping and concurrency APIs.

New ‘Type’ Property on All Types

All types now have a `Type` property. This property evaluates to the type of the object. Use this property to access properties and methods on the type itself. In previous releases, some of these type system properties and methods were mirrored on objects rather than their types. Those are now deprecated, although they still work in this

release. Instead, access those methods and properties on the type itself. For example, instead of `object.TypeProperty`, use `object.Type.TypeProperty`.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Find deprecated references to type properties and change your code to access those properties directly on the type, not the original object.

Exception Changes If No Current User and Creating New Bundle

If there is no current user (a batch process or startable plugin) and you call the `runWithNewBundle` method with no user argument, Gosu throws an exception. However, the behavior changed in ClaimCenter 6.0 in an important way. Instead of throwing the exception during the commit phase (after your code runs), Gosu throws the exception immediately before your Gosu block runs.

This only affects unusual usages of this API when there is no user context. This change does not affect the typical case of calling `runWithNewBundle` method when there is already a user context, such as in rule set execution or PCF files. The behavior is the same as before. The `runWithNewBundle` method uses the current user.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: Review all uses of the run with new bundle API. In the rare case that you use this API in code with a current user, it throws an exception before your block runs. Ensure your code does not rely on the old behavior in this unusual case.

Block Declarations Now Require Argument Names

Gosu block declarations now require argument names.

For example, this used to be a valid block declaration:

```
var x( String ) : String
```

Change the declaration to include a variable name for the argument:

```
var x( s : String ) : String
```

This affects declarations of variables but also affects method declarations in which one parameter of the method is a block.

If Studio flags a compiler error on such code, fix the error by adding a variable name for each parameter.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Fix any compile errors related to changes in block declarations.

Function Pointers and Nested Functions Now Unsupported

Gosu function pointers are no longer supported.

For example, this used to be valid:

```
// assign to this variable the value of the built-in Gosu global function "print",
// which takes a String argument and returns void. Gosu permits this type of assignment.
var myPrint = print
myPrint( "hello typesafe function pointers" )
```

This is an unusual pattern to use. Some customers might have used this for callbacks, such as saving a reference to a static method.

Instead, save a reference to a block that calls the static method or whatever other logic you need.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Fix any compile errors related to changes in function pointers.

Nested Functions Disallowed

Gosu does not support functions within functions. (Although Gosu does support the feature called Gosu blocks).

Studio now generates compile errors if you attempt functions within functions such as this:

```
static function doAction(messageContext : entity.MessageContext, actions : gw.rules.Action) {
    foo()
    function foo() {
    }
}
```

The main impact for customers is that business rules cannot define custom functions. Rules internally become functions and so the functions within functions rule applies.

Annotation Syntax Change for Run Time Access

The syntax changed for accessing Gosu annotations and properties at run time.

The old syntax is as follows, given a type called MyType and annotation class MyAnnotation.

```
var annotations = MyType.TypeInfo.getAnnotation(MyAnnotation)
for (a in annotations) {
    print(" Author: " + a.MyProperty);
}
```

The new syntax is:

```
var annotations = MyType.Type.TypeInfo.getAnnotation(MyAnnotation)
for (a in annotations) {
    print(" Author: " + (a.Value as MyAnnotation).MyProperty);
}
```

UPGRADE TASK

Priority: Before starting the server

Summary: Fix the syntax of any access of annotation data at run time.

New and Changed in System Administration in 6.0

New in System Administration in 6.0

This topic describes what is new for system administration of ClaimCenter 6.0. This includes:

- Batch Process to Purge Completed Workflows
- Graph Validation Checks

Batch Process to Purge Completed Workflows

Each time activities are created as part of the claim workflow, they are added to the `cc_workflow`, `cc_workflowlog` and `cc_workflowworkitem` tables. Once the activities are completed, their table entries are marked as such and are never used again. These tables grow in size over time and can adversely affect performance as well as waste disk space. Excessive records in these tables also negatively impacts the performance of the database upgrade.

ClaimCenter 6.0 includes a batch process to purge completed workflows and their logs that are older than a configurable number of days. Guidewire recommends that you purge completed workflows and their logs periodically. This reduces the problem of a large number of workflow log records causing performance issues.

You can also purge only the logs associated with completed workflows older than a certain number of days. This process leaves the workflow records and removes only the workflow log records.

For more information, see “Purging Old Workflows and Workflow Logs” on page 57 in the *System Administration Guide*.

Graph Validation Checks

ClaimCenter uses a domain graph and an admin graph for features such as archiving and purging of claims. These graphs are constructed from the ClaimCenter data model. It is possible to configure your data model in

such a way that these graphs are not usable. In ClaimCenter 5.0, the graph validation checks were only run if the `ArchiveEnabled` parameter was set to `true` in `config.xml`. In ClaimCenter 6.0, the graph validation checks always run. Some of these checks will prevent the server from starting if they fail.

For more information, see “Graph Validation Checks” on page 64 in the *System Administration Guide*.

Changed in System Administration in 6.0

This topic describes what has changed for system administration of ClaimCenter 6.0. This includes:

- Application Server Requirements
- Database Server Requirements
- Operating System Requirements
- Java and ANT Versions
- Caching Mechanism
- Administrative Data Export Granularity

Application Server Requirements

Guidewire has updated the application server requirements for this release. See “Configuring the Application Server” on page 13 in the *Installation Guide* for supported application server versions.

Database Server Requirements

Guidewire has updated the database server requirements for this release. See “Configuring the Database” on page 20 in the *Installation Guide* for supported database versions.

Operating System Requirements

Guidewire has updated the operating system requirements for end-user clients and development workstations. See “Client Information” on page 27 in the *Installation Guide* and “Development Workstation Information” on page 27 in the *Installation Guide*.

Java and ANT Versions

Guidewire has updated the supported Java and ANT versions. See “Development Workstation Information” on page 27 in the *Installation Guide* for updated Java and ANT versions for a development environment. See “Configuring the Application Server” on page 13 in the *Installation Guide* for updated Java versions for a production environment.

Caching Mechanism

ClaimCenter uses a single cache at the application server level rather than the per-object cache used in previous versions. The configuration of the cache is therefore simpler. Guidewire has removed parameters related to the per-object caching model, including:

- `CheckCacheSize`
- `ClaimAccessCacheSize`
- `ClaimCacheSize`
- `ClaimContactCacheSize`
- `ClaimContactRoleCacheSize`
- `ClaimStaleTimeSeconds`
- `ContactCacheSize`

- DefaultCacheSize
- DefaultEvictTimeMinutes
- DefaultStaleTimeSeconds
- GroupUserCacheSize
- TAccountCacheSize
- UpgradeRowCountCacheSize
- UpgradeStepCacheSize
- UpgradeTableRegistryCacheSize
- UpgradeVersionTriggerCacheSize
- UpgradeVTDBMSDumpCacheSize
- UpgradeVTStatementCacheSize
- UpgradeDBParameterPairCacheSize
- UpgradeDBParameterRowCacheSize
- UpgradeDBParameterSetCacheSize
- UpgradeDBStorageSetCacheSize
- UpgradeDBStorageSetColumnCacheSize
- UpgradeDBStorageSetResultCacheSize
- UpgradePhaseCacheSize
- UserCacheSize

See “Understanding Application Server Caching” on page 70 in the *System Administration Guide* for information on how to configure the application server cache for ClaimCenter 6.0.8.

Administrative Data Export Granularity

ClaimCenter 6.0 provides a finer level of granularity for exporting administrative data, so you can limit exports to the administrative data that you are interested in. See “Importing and Exporting Administrative Data from ClaimCenter” on page 121 in the *System Administration Guide*.

New and Changed in Integration in 6.0

New in Integration in 6.0

This topic describes what is new for integration with ClaimCenter 6.0. This includes:

- “Tools for XML Export of Types for Integration” on page 81
- “Asynchronous Document Store and Transport” on page 82
- “Startable Plugins” on page 82
- “Create Custom Batch Processes” on page 82
- “Profiler Web Services” on page 82
- “Simple Servlets” on page 82
- “Field-level Encryption” on page 83
- “Claim Search (PolicyCenter Integration)” on page 83
- “Authentication User Role Syncing” on page 83
- “Large Loss Notification and Policy System Notification Architecture” on page 84
- “New Preupdate Handler Plugin” on page 84
- “Backup Withholding Plugin” on page 84

Tools for XML Export of Types for Integration

You can now export an entity to XML using Guidewire-standardized XSD format. Studio includes a visual tool that helps you export business data entities (and other types like Gosu classes) to XML. You can select which properties are required or optional for each integration point. You can export an XSD to describe the data interchange format you selected. Finally, you can write Event Fired rules that extract the specified properties from the entity and generate an XML payload that conforms to your custom XSD. You can also use this tool to generate XML and XSDs for Gosu classes and other types, not just entities. For details, see “The Guidewire XML (GX)

Modeler” on page 262.

Asynchronous Document Store and Transport

Some document production systems generate documents slowly. When many users try to generate documents at the same time, multiple CPU threads compete and that makes the process even slower. One alternative is to create documents asynchronously so that user interaction with the application does not block waiting for document production. Similarly, **transfer** of large documents through the application server to an external document storage system typically requires significant memory, resources, and real-world time. Even in the best case scenario of available memory and CPU resources on the server, an external document storage system or the network between the servers may be slow. In that case, synchronous actions with large documents may seem unresponsive to a web user.

To address these issues, ClaimCenter provides a system to asynchronously **send** document to your document management system. This allows an asynchronous send request to return immediately without bringing documents into application server memory. A separate thread on the batch server sends documents to your real document management system using the messaging system. For more information, refer to “Asynchronous Document Storage” on page 276 in the *Integration Guide*.

Startable Plugins

You can register custom code that runs at server startup in the form of a *startable plugin* implementation. You can use this type of plugin as a listener, such as listening to a JMS queue. You can selectively start or stop each startable plugin in an administrative interface, unlike other types of plugins. For more information, see “Startable Plugins” on page 128 in the *Integration Guide*.

Create Custom Batch Processes

You can now write your own batch processes. For more information, see “Custom Batch Processes” on page 325 in the *Integration Guide*.

There are several related changes:

- The data model file `tl_pl_system.xml` used to contain all `BatchProcessType` typecodes for all products. That moved into the `tl_cc_system.xml`. The new typelist file `BatchProcessType.xml` now defines the list of batch processes. You can modify this typelist in Studio to create a new batch process.
- For `scheduler-config.xml`, there is an upgrader tool that replaces the old-style batch process name strings with the new batch process type.
- There is a new plugin interface called `ProcessesPlugin` that helps instantiate new batch processes to support this new feature for custom batch processes. For more information, see “Custom Batch Processes” on page 325 in the *Integration Guide*.

Profiler Web Services

The new `IProfilerAPI` web service interface controls the ClaimCenter profiler from an external system.

Simple Servlets

You can define simple web servlets inside your ClaimCenter application. You can define extremely simple HTTP entry points to custom code using this approach.

These are separate from web services that use the SOAP protocol. These are separate from Guidewire PCF entry-points. Instead, you can define arbitrary code triggered from any URL as long as you can define a Gosu block that can determine from the URL whether it owns the request. There is no complex object serialization or deserialization such as is done in the SOAP protocol.

The implementation uses the standard Java classes in the package `javax.servlet.http` to define the servlet request and response. For more information, see “Servlets” on page 97 in the *Integration Guide*.

Field-level Encryption

ClaimCenter now supports field-level encryption. For more information, see “Encryption Integration” on page 401.

Claim Search (PolicyCenter Integration)

If you use a supported version of PolicyCenter, PolicyCenter uses the new ClaimCenter web service `PCCClaimSearchIntegrationAPI` web service to retrieve claim summaries based on search criteria. Typically you do not need to call this directly, since PolicyCenter calls this automatically if you configure PolicyCenter to connect to ClaimCenter for claim search.

If you use a policy system other than PolicyCenter, your policy system can call this web service. However, this web service is written specifically to support a data model very similar to what PolicyCenter needs. If you use a policy system other than PolicyCenter, consider writing your own custom web services that directly address integration points between ClaimCenter and your specific policy system.

For more information, see “Claim Search Web Service For Policy System Integration” on page 454 in the *Integration Guide*.

Authentication User Role Syncing

Many customers use an external authentication system, such as an LDAP directory. If your external directory controls the most up-to-date version of user roles (LDAP system roles), you can now synchronize ClaimCenter user roles at ClaimCenter logon time.

New Configuration Parameter for User Syncing

There is a new configuration parameter `ShouldSynchUserRolesInLDAP` in `config.xml`. If its value is `true`, the application synchronizes contacts with the roles they belong to after authenticating with the external authentication source.

Changes to the Authentication Plugin Callback Class

Even if you set the `ShouldSynchUserRolesInLDAP` parameter to `false` (or you do not set it), you can synchronize user roles after authenticating.

The plugin interface `AuthenticationServicePlugin` plugin interface has a method called `setCallback`, which ClaimCenter calls and provides an object called an authentication handler. You can call a few utility methods on this object, documented in “User Authentication Service Plugin” on page 242 in the *Integration Guide*. Your plugin implementation must save a private instance variable reference to it so it can use it later.

In this release, the plugin interface itself did not change. However, there is now an extra method to synchronize user roles. After you get the list of roles from the external system, put the user name and the list of roles in an `AuthenticationResponse` object. Next, pass that authentication response object to the `synchUserRoles` method. Note that there is a constructor for the `AuthenticationResponse` object that takes a login ID and a role list. For details, see “User Authentication Service Plugin” on page 242 in the *Integration Guide*.

Alternative (or Supplemental) User Role Syncing

The user role synchronization as part of ClaimCenter authentication does not provide absolute protection for changes to user roles. Alternately, you might consider a separate system to push changes from the authentication system directly into ClaimCenter as soon as changes happen in that system. For example, if someone’s role’s change and the user is currently logged into ClaimCenter, it might be important to reflect that change immediately. To do this, write your own ClaimCenter web service that your authentication system calls to update the

roles. Because user roles are checked at a low level in every database commit, business data always follows the security model logic.

You could provide one of these user role synchronization systems, or both, depending on your own security process and needs.

Large Loss Notification and Policy System Notification Architecture

ClaimCenter can now notify a policy system if a claim reaches a critical threshold, defined by policy type. The policy system can take appropriate actions to notify the policy's underwriter. When the claim hits the threshold, ClaimCenter creates a message. A special message transport plugin for this purpose sends the message to a policy system to send the claim's policy number, loss date, and the total gross incurred.

If you also use PolicyCenter, ClaimCenter notifies PolicyCenter. A built-in version of this large loss notification service calls web services that PolicyCenter exposes. The PolicyCenter web service creates an `Activity` on the policy's `Account` and adds a referral reason (`ReferralReason`) to the policy.

ClaimCenter includes a new policy system notification architecture to support this feature. You can use this new architecture to add easily other types of policy system notifications.

By default, ClaimCenter does not enable large loss integration to PolicyCenter. To enable it or for more information about the architecture of this system, refer to "Policy System Notifications" on page 443 in the *Integration Guide*.

As a consequence of this change, there is a new plugin called `IPolicySystemNotificationPlugin` and the `IPolicyRenewalAlert` plugin is now deprecated. If you used a policy renewal alert for large loss notification, use the built-in support for that and add your integration code to the `IPolicySystemNotificationPlugin` implementation. If you want to add entirely new types of notification to your policy system, add a new notification type. Then, add your actual call to the external system into your `IPolicySystemNotificationPlugin` implementation. Your methods will not be in the plugin interface, but ClaimCenter includes another type of registration of your new notification type. See "Policy System Notifications" on page 443 in the *Integration Guide* for details.

New Preupdate Handler Plugin

If you want to implement your preupdate handling in plugin code rather than in the built-in rules engine, register an implementation of the `IPreUpdateHandler` plugin interface. For details, see "Preupdate Handler Plugin" on page 337 in the *Integration Guide*.

Backup Withholding Plugin

In previous releases, you could customize deductions with `IDeductionAdapter`. For backup withholding for checks, there is now an additional plugin interface called `IBackupWithholdingPlugin`.

This is similar to the `IBackupWithholdingPlugin` plugin interface. Differences include:

- The `IDeductionAdapter` handles deductions in a generic way, rather than for backup withholding for checks.
- `IDeductionAdapter` requires use of special template to generate parameters into a large `String` data object. In contrast, the `IBackupWithholding` plugin takes a typesafe `Claim` object for its method parameter.

The built-in implementation of the `IBackupWithholdingPlugin` plugin uses a utility class that does all the work of calculating deductions. You can view and edit the code that implements the built-in behavior of this plugin. In studio, refer to the Gosu class `gw.util.BackupWithholdingCalculator`.

Note: The default implementation of the `IDeductionAdapter` plugin simply calls the registered version of the `IBackupWithholding` plugin interface. In turn, that class calls the backup withholding utility class `gw.util.BackupWithholdingCalculator` to do all of its work.

For details, see "Deduction Calculations for Checks" on page 235 in the *Integration Guide*.

Changes in Integration in 6.0

This topic describes what has changed for integration with ClaimCenter 6.0. This includes:

- “Changes to Structure of Integration-related Files and Scripts” on page 85
- “Copy the Plugin Entity Library to Your Configuration Module” on page 86
- “ISO Changes in ClaimCenter 6.0” on page 87
- “FNOL Mapper is Now a Server-Side Tool” on page 94
- “Financials Integration APIs for Acknowledgement-based Transitions” on page 95
- “Validation Plugin Removed” on page 96
- “Multicurrency-related Changes” on page 97
- “Claim Financials Web Service API Changes” on page 98
- “Messaging API Changes” on page 98
- “Document Management Plugin Implementation Location Changes” on page 101
- “Parameterization of Types Stripped from Java External Entities” on page 101
- “New Plugin Registry (Old Version Deprecated in Gosu)” on page 101
- “IMessagingToolsAPI Web Service Interface Changes” on page 102
- “IDataExtractionAPI Web Service Interface Deprecated” on page 102
- “IClaimAPI Web Service Interface Changes” on page 103
- “IClaimFinancialsAPI Web Service Interface Changes” on page 103
- “Document Management Plugin Changes for Availability” on page 103
- “Geocoding Plugin Changes for Map Options” on page 104
- “Summary of Plugin Changes, Additions, and Removals” on page 104
- “Web Service Changes, Additions, and Removals” on page 105
- “Authentication Integration Changes” on page 106
- “Enforcement of Web Service Type Name Conflicts” on page 106
- “Metropolitan Timeout Changes” on page 106

Changes to Structure of Integration-related Files and Scripts

The structure of the configuration environment changed in ClaimCenter 6.0. The following list summarizes the changes:

- The command-line tools have changed completely and are now implemented in Gosu. These command line tools are in the `ClaimCenter/admin/bin` directory. The new command-line implementation of these administrative Gosu tools in the `ClaimCenter/admin` directory. These Gosu tools do **not** require regeneration of any libraries, unlike command line tools in previous versions of ClaimCenter.
- The location of the generated Java libraries and related documentation changed.

The new location for the main documentation for Java development is:

`ClaimCenter/java-api/doc/api`

There is another documentation directory at `ClaimCenter/java-api/doc/gw-cc-plugin`, which documents additional utility classes.

The new location for the libraries is:

`ClaimCenter/java-api/lib`

IMPORTANT Compile your Java code against the files in `ClaimCenter/java-api/lib/`. Those are the only supported versions of Guidewire Java libraries, including the generated files from Guidewire entities. Do not rely on Java libraries in internal modules or in other directories in the product hierarchy. However, you must copy the `gw-entity-cc.jar` file to your configuration directory, as discussed in “Copy the Plugin Entity Library to Your Configuration Module” on page 86.

- The location of the generated SOAP libraries and related documentation changed.

The new location for the main SOAP client documentation is:

`ClaimCenter/soap-api/doc/api`

There is another documentation directory at `ClaimCenter/soap-api/doc/gw-cc-plugin`, which documents additional utility classes.

The new location for the libraries is:

`ClaimCenter/soap-api/lib`

IMPORTANT Compile your Java SOAP client code against the files in `ClaimCenter/soap-api/lib/`. Those are the only supported versions of Guidewire Java libraries, including the generated files from Guidewire entities. Do not rely on Java libraries in internal modules or in other directories in the product hierarchy.

- The targets (the parameters) for the `gwcc.bat` file changed. The `regen-toolkit` target no longer works. Do not rely on it. The work of the `regen-toolkit` target now happens in two separate new targets:
 - `regen-java-api` - regenerates libraries and documentation related to Java plugin development.
 - `regen-soap-api` - regenerates libraries and documentation related to web services (SOAP) development.
- Because of the data that moved to new folders, the `ClaimCenter/toolkit` directory no longer exists. Java and SOAP documentation are in separate folders, there is no longer a `index.html` file that points to both types of generated documentation. This file was at the root of the `ClaimCenter/toolkit` directory.

IMPORTANT The *toolkit* as it existed in previous releases no longer exists. However, the library files and documentation that used to be in those directories still exist. Instead, refer to the `java-api` and `soap-api` subdirectories. Additionally, the `regen-toolkit` target is gone, replaced by `regen-java-api` and `regen-soap-api`.

On a related note, the Data Dictionary and Security Dictionary are also in different locations. They are now inside the `ClaimCenter/build/dictionary` directory.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Update any tools or scripts in your build system to use the new locations of scripts and Java libraries.

Copy the Plugin Entity Library to Your Configuration Module

Starting in ClaimCenter 6.0, if your Java plugin uses Guidewire entities at all, you must also copy the Java entity library file `gw-entity-cc.jar` to the configuration module shared libraries folder. This is required for ClaimCenter to find the library at runtime.

In other words, you must copy the following file:

`ClaimCenter/java-api/lib/gw-entity-cc.jar`

Copy it to the following location:

```
ClaimCenter/modules/configuration/plugins/shared/lib/gw-entity-cc.jar
```

If you did not regenerate the Java API libraries since your last change to the data model, it might be necessary to regenerate them. It might not be necessary, depending on what kinds of changes you made. For details, see “Regenerating the Integration Libraries” on page 17 in the *Integration Guide*.

IMPORTANT If you use Java plugins or other Java code that uses entities, you **must** manually copy the entity libraries file `gw-entity-cc.jar` to your configuration module every time after regenerating it.

Remember to update your development flow to accommodate this change.

✓ UPGRADE TASK

Priority: Before starting up the server

Summary: Update your development build system to copy the entity libraries JAR file to your configuration module after regenerating it.

ISO Changes in ClaimCenter 6.0

The implementation of the integration for Insurance Services Office changed substantially in ClaimCenter 6.0. The following subtopics describe the important changes.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: Decide whether you want to use new optional claim-based ISO sending. Also, review all ISO code for changes in this release and update your code to use the new payload generation classes. It is possible to use your old payload generation code in this release, but it is best to convert your code to the new system.

Optional Claim-based ISO Sending (Not Just Exposures)

In ClaimCenter 5.0, what ISO called claims corresponded to an exposure in ClaimCenter. In other words, ClaimCenter sent individual exposures to ISO but did not send entire ClaimCenter claims. In ClaimCenter 6.0, the ISO integration can optionally implement the ISO integration as claim-based integration or exposure-based. This is controlled by the `ISO.properties` file property `ClaimLevelMessaging`. You can keep using only exposure-based ISO support if you prefer.

Using the claim-level ISO integration, which is the new default, the claim summary page displays a button that re-sends a claim to ISO or check on the ISO status. Similarly, the exposure-level ISO information disappears from the user interface. As in ClaimCenter 5.0, if you use exposure-based ISO messaging, ClaimCenter displays the ISO resend button on the exposure pages.

If you switch from exposure-based ISO messaging to claim-based ISO messaging, ISO continues to track your old submissions of ClaimCenter exposures in the ISO system. ClaimCenter continues to maintain the following properties:

- `exposure.ISOSendDate` - the ISO send date, which is set in the same database transaction as the new message to ISO)
- `exposure.KnownToISO` - the Boolean flag that indicates whether ISO has acknowledged the message. This property is set in the message acknowledgement database transaction

If ClaimCenter attempted to send an exposure to ISO (that is, the `ISOSendDate` is non-null), ClaimCenter handles ISO replies on that claim only using exposure-based mode. Once a user attempts to send ISO a claim or exposure, that claim is fixed in that mode (claim-based or exposure-based). This allows a smooth transition from

exposure-based ISO messaging to claim-based ISO messaging while still handling exposures submitted to ISO before the switch to claim-based messaging.

The actual implementation of this algorithm is in a new property called `ISOClaimLevelMessaging` defined on claims as a Gosu enhancement. It returns `true` to tell ClaimCenter to use claim level messaging for this claim. It returns `true` to tell ClaimCenter to use claim level messaging for this claim. It returns `false` to indicate exposure-based messaging. The built-in implementation uses the algorithm described in the previous paragraph. However, you can configure this property for special business needs. For example, you could choose to send messages at the claim level for a particular loss type or line of business. To make modifications like this, edit the `GWClaimISOEnhancement` enhancement file. However, if you change the logic of the `ISOClaimLevelMessaging` property, you must follow these rules:

- The return result must be deterministic given the same claim. In other words, it must return the same result for that claim if called multiple times.
- However, once ClaimCenter sends a claim or exposure to ISO (the `ISOEndDate` is non-null), that claim must always use that same mode from then on. You cannot mix both ISO levels on a single claim. Even if you make other changes, the return result must preserve this logic.

Many of the properties and methods on exposures related to ISO now also apply to claims. From an implementation perspective, the similarities between exposures and claims are defined by the fact that `Claim` and `Exposure` entities both implement the new interface called `ISOReportable`. Be aware of this important change as you review the documentation or you are looking through built-in code such as payload generation Gosu code. Where you see a reference to `ISOReportable`, you can usually think of this as “a claim or an exposure”. However, whether ClaimCenter actually uses claim-level messaging is controlled primarily by the `ISOClaimLevelMessaging` property, which is dynamic as discussed earlier in this topic.

Because you can send either claims or exposures to ISO, there are now two different types of ISO reports. This required a data model change for the `ISOMatchReport` entity to be two separate entities: `ClaimISOMatchReport` for claims and `ExposureISOMatchReport` for exposures. During upgrade, ClaimCenter converts any entities of type `ISOMatchReport` to be `ExposureISOMatchReport` entities.

The `ISOMatchReport` entity no longer exists. However, `ISOMatchReport` is now a delegate that defines the interface for ISO match reports. The claim and exposure versions of ISO match reports implement this interface. Both `ClaimISOMatchReport` and `ExposureISOMatchReport` entities now contain the properties `ISOClaim` and `ISOExposure`. For `ClaimISOMatchReport`, the `ISOClaim` property points to the claim and the `ISOExposure` property is `null`. For `ExposureISOMatchReport`, the `ISOExposure` property points to the exposure and the `ISOClaim` property points to the claim that owns that exposure.

If you are customizing the ISO match report code, it might be useful to know the API for creating a new match report. From a claim or an exposure (that is, for all `ISOReportable` entities), to create the correct type of match report subtype, call the `ISOReportable.addNewISOMatchReport()` method.

New Coverage Mapping File

The ISO integration now uses a text file to configure how ClaimCenter coverages map to ISO coverages. A comma-separated values (CSV) file called `ISOCoverageCodeMap.csv` controls this mapping.

ClaimCenter coverage codes represent the policy type, coverage type and coverage subtype. These map to the ISO policy type, coverage type and loss type. In ClaimCenter 5.0 this mapping happened in the `TypeCodeMap.xml` file by mapping each type (policy, coverage, coverage subtype) individually.

The new `ISOCoverageCodeMap.csv` file maps a ClaimCenter policy type, LOB code (optional), coverage type and coverage subtype to an ISO policy type, coverage type and loss type. The format is a comma-delimited row containing the following fields in this order:

- source ClaimCenter policy type
- source optional line of business code (can be blank)
- source coverage type

- source coverage subtype
- ISO policy type
- ISO coverage type
- ISO loss type

For example a couple of lines in this file might be:

```
auto_comm,,ABI,abi_bid,CAPP,BODI,BODI
businessowners,pr,ADPERINJ,adpersinj_gd,CPBO,OTPR,OTPR
```

Notice that the first example line does not include a line of business. The second line includes a line of business.

This new format allows for hierarchy differences between the ClaimCenter and ISO policy type hierarchies, which were not supported by `TypeCodeMap.xml`. For example, ClaimCenter might share the same coverage type, X, between two different policy types, A and B. But ISO might have two different coverage types ISOAX and ISOBX to handle this case. In ClaimCenter 5.0, you could only have one ISO mapping (choose either X → ISOAX or X → ISOBX) and then patch up the other case in rules.

In the new coverage mapping system, you can now do the following:

```
A,,X,CS,ISOA,ISOAX,ISOCS
B,,X,CS,ISOB,ISOBX,ISOCS
```

This maps coverage type X to ISOAX if the policy type is A, but maps ISOBX if the policy type is B.

There are some special features to the `ISOCoverageCodeMap.csv` mapping:

- The ISO coverage code is empty in several mappings. These are usually mappings for first party property claims and exposures; ISO does not use a coverage code for such losses.
- The LOB code. Normally ClaimCenter policy types map straight to ISO policy types, and entries for these policy types omit the `LOBCode` field. But there are some cases in which a single ClaimCenter policy type maps to multiple ISO policy types. In such cases the `LOBCode` can be added to the mapping, to narrow it down to a particular ISO policy type. For example, in the out of box configuration `businessowners,pr` (business owners policy type, property LOB) maps to ISO's CPBO, while `businessowners,g1` maps to ISO's CLB0.
- In some cases the ClaimCenter coverage subtype is not very specific and the claim's loss cause gives a much better sense of the ISO loss type to use. The mapping file allows entries like `LossCause/OTPR` in the ISO loss type column. This tells ClaimCenter to try mapping the claim's loss cause field to an ISO value using the `TypeCodeMap.xml` file. If no mapping is found default to the ISO loss type OTPR. This extended mapping is mainly used for homeowner's at the moment.

The ISO code uses a slightly higher-level lookup operation based on a ClaimCenter policy type, LOB code, coverage type, coverage subtype and loss cause using a two-step full mapping lookup:

1. First ClaimCenter looks for appropriate mapping lines in `ISOCoverageCodeMap.csv`
2. If that lookup returns a loss type of the form `LossCause/OTPR`, ClaimCenter looks up the loss cause in the `TypeCodeMap.xml` mapping.

You can request a full lookup from Gosu using the `ISOTranslate.getCoverageCodes()` method.

Updated ISO Transport and New Payload Generation System

There is an entirely new ISO payload generation system, which is implemented using a new ISO transport, which is a messaging plugin. Some of these changes were designed to more effectively take advantage of the native XML and XSD handling features available in Gosu.

Note: If you previously customized your ISO files significantly, during upgrade you can choose to use the **old system** to minimize upgrade changes during upgrade. For more information, see “Using Your Old ClaimCenter 5.0 ISO Payload Generation and Messaging” on page 91.

In ClaimCenter 5.0, ClaimCenter generated ISO payloads using functions in the ISO library files in Studio within **Classes → Libraries → iso**. This file contains a variety of functions that ClaimCenter called to generate each type of

ISO payload using the `ISOPayloadGenerator` utility class. If you wanted to modify the payloads, built-in XPath utility classes helped you patch the generated XML with additional (or changed) fields.

In ClaimCenter 6.0, this ISO libraries file remains. However, this file is no longer the main location for payload generation code. The new payload generation system uses special new classes, each of which generates payloads for a certain type of ISO *loss section*. A loss section is the ISO term for a type of loss such as a property loss, a vehicle loss, and so forth. To customize how ISO generates the payload XML for a loss section, modify the ClaimCenter class as defined in the following table.

Type of loss	ISO loss section	Customer class to modify in ClaimCenter package gw.api.iso
auto loss	AutoLossInfo	ISOAutoLossSection
information about injured people (standard, including third-party property)	ClaimsInjuredInfo	ISOInjuryLossSection
information about injured people (Workers' Comp)	ClaimsInjuredInfo	ISOWCInjuryLossSection
property loss insurance information for a person or insured party	PropertyLossInfo/ClaimsSubjectInsuranceInfo	ISOPropertyLossSection
property loss insurance information for an object	PropertyLossInfo/ItemInfo	ISOMobileEquipmentLossSection
property loss insurance information for a water craft	PropertyLossInfo/Watercraft	ISOWatercraftLossSection

All of these files in the previous table represent pairs of implementation classes. ClaimCenter organizes files like this to simplify code merges for ISO payload generation during every upgrade. For each section, the implementation class for your changes is the class listed in the previous table. However, these classes extend another base class that contains Guidewire core code for default behaviors. This base file's file name has the suffix "Base". For example, the `ISOPropertyLossSection` class extends from the `ISOPropertyLossSectionBase` class, which contains the bulk of the Guidewire code. If you want to see the existing implementation, refer to the base file. However, if you want to modify the behavior, Guidewire strongly recommends simply overriding methods in the non-base file rather than modifying the base class directly.

The code in the ISO library file determines which loss section class to use by getting the exposure enhancement property called `ISOLossSectionType`. The Gosu enhancement file `GWExposureISOEnhancement` implements this dynamic property. Customize that file if you want to change the mapping of exposure type to loss section type.

Because of these changes, in ClaimCenter 6.0, utility classes that relate only to the ClaimCenter 5.0 ISO implementation are now deprecated in this release. Similarly, the `ISOPayloadGenerator` object is deprecated because it is only useful for using ISO messaging based on ClaimCenter 5.0.

Changes to ISO Reply Handling

In ClaimCenter 6.0 using the new ISO messaging transport, ISO handles replies (message acknowledgements) from ISO using different code compared to previous releases. Using the new style ISO messaging, the replies are handled by the new class `ISOReplyPlugin`, which is an implementation of the `MessageReply` plugin interface. The `MessageReply` interface handles the acknowledgements from asynchronous messaging sending for messages sent from a `MessageTransport` plugin. In the new release, this reply plugin calls out to another class called `ISOReply` to do the important actual work for ISO replies.

Refer to the `ISOReply` class in Studio if you want to look at how ClaimCenter handles replies or if you want to customize reply handling.

Because of this change:

- Do not add any lines for match report property mapping in the `ISO.properties` files. These are now unsupported. Instead, you can customize the `ISOReply` class implementation in the `populateMatchReportFromXML` method. Perform any necessary logic there. For method arguments, this method gets the match report object to populate plus a Gosu object representing the match report XML. You can add as much additional post-reply logic you want.
- ClaimCenter no longer adds activities as part of rule execution. Instead, the `ISOReply` class adds activities as part of the acknowledgement of the message to ISO.

Using Your Old ClaimCenter 5.0 ISO Payload Generation and Messaging

If you previously customized your ISO files significantly, during upgrade you can choose to use the **old system** to minimize upgrade changes during upgrade. To do use your own ISO customizations from ClaimCenter 5.0, perform the following tasks:

1. Copy your `libraries.iso` file from your ClaimCenter 5.0 configuration, including any customizations, into your ClaimCenter 6.0 configuration.
2. In Studio, click Resources → Messaging. Next, click on the line labeled `isoTransport`.
3. Change the implementation class for `isoTransport`. Change the path to the same messaging transport plugin implementation class used in ClaimCenter 5.0:
`com.guidewire.cc.system.integration.messaging.iso.ISOMessageTransport`.
4. Copy your ISO-related rules from a previous release. However, you **must** remove any ISO rules code that creates activities. ClaimCenter ISO code now handles this in the reply handling. This new approach is a more consistent way to handle activities with respect to database transactions (bundles) for replies. Perform the following steps:
 - a. In Studio, go to Configuration → Event Message → Event Fired. Next, in the Rules pane, navigate to ISO → Exposure → Exposure Change.

- b. Remove the following code from the ClaimCenter 5.0 built-in ISO rules that you copied to ClaimCenter 6.0 configuration environment:

```
if (libraries.ISO.hasNewMatchReports(exposure)) {
    exposure.Claim.createActivityFromPattern( exposure,
        ActivityPattern.finder.getActivityPatternByCode("iso_matches"))
}
var newError = libraries.ISO.getNewErrorMessage( exposure )
if (newError != null) {
    var activity = exposure.Claim.createActivityFromPattern( exposure,
        ActivityPattern.finder.getActivityPatternByCode("iso_matches"))
    activity.Subject = "ISO returned error response"
    activity.Description = newError
```

These lines add new activities if ClaimCenter created new match reports on an exposure, or if a new ISO error message occurs.

ClaimCenter now creates these activities in the Gosu class `ISOReply`. That class extends `ISOReplyBase`, and that class creates activities in the `updateReportableOnSuccess` and `updateReportableOnErrorStatus` methods.

Multiple Role Mapping for Claim Parties

The `ISO.properties` file has two contact-related sections. One section maps claim party codes to a `ClaimContact` entity role typecode. For example:

```
ISOClaimParty.BS = repairshop
```

The code on the left is the ISO claim party code, and the code on the right of the equals sign is the ClaimCenter `ClaimContact` entity role typecode. Add or change these mappings as needed for any changed or added contact role typecodes.

In ClaimCenter 5.0, each line can map to only one ClaimCenter typecode.

In ClaimCenter 6.0, each line can map to one or more ClaimCenter typecode. To include more than one, separate the role typecodes with commas.

Theft Changes

In ClaimCenter 5.0, theft was considered a separate loss type. However, it is actually similar to a property loss. In ClaimCenter 6.0, ClaimCenter uses a property loss section by default and sets a few extra fields as needed, such as `Exposure.LostPropertyType`.

Agency ID Now Optionally at Claim Level

In ClaimCenter 5.0, the ISO integration only used a single ISO Agency ID, which is like a user name for your ISO account.

In ClaimCenter 6.0, you can optionally override the agency ID by claim with the `AgencyID` property on a claim. If that field is null, ISO uses the standard Agency ID property defined in `ISO.properties`.

Field Mapping Additions

New mappings for Auto & Watercraft Recovery / Salvage

ISO Aggregate	ISO Tag	CC Entity	CC Field
InvestigationInfo	RecoveryInfo		
InvestigationInfo	SalvageInfo		
RecoveryInfo	ItemRef	N/A	ID of Vehicle (AutoLossInfo)
RecoveryInfo	RecoveryAgencyRef	N/A	ID of Recovery Agency (ClaimParty)
RecoveryInfo	RecoveryStatusCd	MobilePropertyIncident	RecoveryCondition
RecoveryInfo	RecoveryDt	VehicleIncident	DateVehicleRecovered
RecoveryInfo	Addr	MobilePropertyIncident	RecoveryLocation
SalvageInfo	ItemRef	N/A	ID of Vehicle (AutoLossInfo)
SalvageInfo	SalvageAgencyRef	VehicleIncident	SalvageCompany
SalvageInfo	ItemValueReceivedAmt	VehicleIncident	SalvageProceeds
SalvageInfo	OwnerRetainingSalvageInd	VehicleIncident	OwnerRetainingSalvage

New Fields

There are new ISO optional fields to capture more ISO requested information. Refer to the following table for more information.

Entity	Field	Type	Description	ISO equivalent
VehicleIncident	HitAndRun	Boolean	To indicate if a claim involves hit and run	com.iso_HitAndRunInd
VehicleIncident	PhantomVehicle	Boolean	Unknown 3rd party vehicle involved, such as Hit and Run	com.iso_PhantomVehInd
FixedPropertyIncident	Arson	Boolean	To indicate if a claim involves arson	IncendiaryFireInd

Entity	Field	Type	Description	ISO equivalent
ContentIncident	LocationOfTheft	Typelist	To describe the Location where the property was stolen	TheftLocationCd
FixedPropertyIncident	OccupancyType	Typelist	To describe where the property in question is occupied	OccupancyTypeCd
VehicleIncident	RecoveryCondition	Typelist	Typelist	RecoveryInfo.RecoveryStatusCd
VehicleIncident	RecoveryLocation	Address	The Address at which the recovery was made	RecoveryInfo.Addr
VehicleIncident	OwnerRetainingSalvage	Boolean	Owner will retain the salvaged car	SalvageInfo.OwnerRetainingSalvageInd
InjuryIncident	AmbulanceUsed	Boolean	True if an ambulance arrived during the loss	CLAIMSINJUREDINFO.com.iso_AmbulanceUsedInd
InjuryIncident	DisabledDueToAccident	Typelist	For non-WC, to characterize the disability	CLAIMSINJUREDINFO.com.iso_DisabledDueToAccidentInd
Policy	AssignedRisk	Boolean	The policy is an Assigned risk from the state	AssignedRiskInd
ContactRole role	RepairShop	Contact Type	ClaimParty role	BS
ContactRole role	Witness	Contact Type	ClaimParty role	WT
ContactRole role	Doctor	Contact Type	ClaimParty role	MD
ContactRole role	Hospital	Contact Type	ClaimParty role	MF

For the five typelists above the typecodes are as follows:

VehicleType:

AT – All Terrain Vehicle (ATV)
 BT – Boat
 CE – Construction Equipment
 FE – Farm Equipment
 HD – Heavy Duty Truck
 LT – Light Duty Truck
 MC – Motorcycle
 MP – Multi Purpose Vehicle
 PC – Passenger Car
 PT – Part
 SB – Snowmobile
 TK – Truck
 TL – Trailer

LocationOfTheft:

R – Residential
 C – Commercial
 O – Off Premises

OccupancyType:

V – Vacant
 U – Under Construction
 O – Occupied (Not an ISO code)

DisabledDueToAccident:

P-Partially Disabled
 T = Totally Disabled
 N = Not Disabled(Not an ISO code)

RecoveryCondition:

0 - Unknown
 2 - No Apparent Damage
 3 - Stripped
 4 - Wrecked
 5 - Burned
 6 - Flood
 7 - Vandalized
 8 - Stripped & Wrecked
 9 - Stripped & Burned
 A - Stripped & Flood
 B - Stripped & Vandalized
 C - Wrecked & Burned
 D - Wrecked & Vandalized
 E - Burned & Vandalized
 F - Flood & Vandalized

FNOL Mapper is Now a Server-Side Tool

The implementation of the FNOL Mapper tool changed significantly. This tool takes an XML file that represents a claim's First Notice of Loss (FNOL) and generates a claim in ClaimCenter. ClaimCenter includes a built-in mapping system for the industry-standard ACORD format. Because customers use the ACORD format slightly differently and might have custom exposure types or additional data model extensions in ClaimCenter, the FNOL mapper tool lets you customize the mapping. The XML file can be in any XML format, and you can create additional mapping classes that help to parse your XML data and populate a claim and its subobjects.

In ClaimCenter 5.0, FNOL mapper was primarily a client-side tool. In ClaimCenter 6.0, FNOL mapper is primarily a server-side tool. The following table compares the old and new versions:

Version	Where mapping happens	Data sent across the wire	Description
ClaimCenter 5.0	Client-side	Deserialized SOAP entity for Claim and its subobjects	First, you configure files on a computer containing the ClaimCenter command line tools and that can access your ClaimCenter server over the network. Next, you run a command line tool that maps the XML file to a ClaimCenter claim. Finally, it sends the claim to ClaimCenter. All the mapping happens in the command line tool. Internally, the tool called the ClaimCenter web service IClaimAPI method addFNOL with a Claim SOAP entity. That version used XPath expressions to customize the mappings.
ClaimCenter 6.0	Server-side	Raw XML in the ACORD format or other custom XML formats that describe FNOL.	First, customize the mapping files within Studio, which are Gosu classes. Next, upload the XML data for a new FNOL using two new methods on the IClaimAPI web service interface. Use <code>importAcordClaimFromXML</code> for ACORD data. Use <code>importClaimFromXML</code> for other XML formats that describe a First Notice of Loss. For custom XML formats, write custom mapping classes to populate a claim and its subobjects. All XML processing happens in the Gosu language, which has native XML processing. You never need to parse raw XML.

For details, see “FNOL Mapper” on page 381 in the *Integration Guide*.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: Convert uses of the client-side FNOL mapper to the new server-side FNOL mapper system.

Financials Integration APIs for Acknowledgement-based Transitions

In ClaimCenter 5.0, some financials status transitions complete only if you marked the message at message creation time using methods on Message entity, such as `message.submittingCheck(Check)`. Then, message acknowledgment automatically triggered the status change.

In ClaimCenter 6.0, the recommended approach is to call new methods on financials objects at message acknowledgment time in your messaging plugins. The transition completes as a result of calling these financials domain methods. Do not use the message-based methods, which are now deprecated.

The following methods are new:

- `Check.acknowledgeSubmission()`
- `Check.acknowledgeTransfer()`
- `Payment.acknowledgeRecode()`
- `Recovery.acknowledgeRecode()`
- `Recovery.acknowledgeTransfer()`
- `Recovery.acknowledgeVoid()`
- `BulkInvoice.acknowledgeSubmission()`
- `Transaction.acknowledgeSubmission()` -- for all transaction subtypes.

If you use the new methods, remove your references to the deprecated message methods (do not mix the styles).

The following Message methods are deprecated:

- `submittingCheck(Check)`
- `transferringCheck(Check)`
- `voidingRecovery(Recovery)`
- `recodingPayment(Payment)`
- `submittingBulkInvoice(BulkInvoice)`
- `submittingTransaction(Transaction)` - for all transaction subtypes

The following table lists the deprecated method and the method that replaces it

Action	Deprecated ClaimCenter 5.0 API (for Event Fired Rules at message creation time)	ClaimCenter 6.0 API (for messaging plugins at message acknowledgement time)	ClaimCenter 6.0 API Description
Submit a check	<code>message.submittingCheck(Check)</code>	<code>check.acknowledgeSubmission()</code>	Updates the check's status to requested if it was requesting, or issued if it was notifying. Updates its payments to submitted. Throws an exception if this check is not in requesting or notifying status
Transfer a check	<code>message.transferringCheck(check)</code>	<code>check.acknowledgeTransfer()</code>	Updates the check's status to transferred. Updates its pendingtransfer payments to transferred. For each transferred payment, updates its onset and offset to submitted. Throws an exception if the check is not in pendingtransfer status.

Action	Deprecated ClaimCenter 5.0 API (for Event Fired Rules at message creation time)	ClaimCenter 6.0 API (for messaging plugins at message acknowledgement time)	ClaimCenter 6.0 API Description
Void a recovery	<code>message.voidingRecovery(recovery)</code>	<code>recovery.acknowledgeVoid()</code>	Acknowledges a message that this recovery was voided. Updates its status to voided. Throws an exception if the recovery is not in pendingvoid status.
Recode a payment	<code>message.recodingPayment(payment)</code>	<code>payment.acknowledgeRecode()</code>	Acknowledges a message that this payment was recoded. Updates its status to recoded. Updates its onset and offset to submitted.
Submit a reserve, recovery, or recovery reserve	<code>message.submittingTransaction(transaction)</code>	<code>transaction.acknowledgeSubmission()</code>	Acknowledges a message that this transaction was submitted. Updates its status to submitted. Throws an exception if this transaction is not in submitting status.
Submit a bulk invoice	<code>message.submittingBulkInvoice(bulkInvoice)</code>	<code>bulkInvoice.acknowledgeSubmission()</code>	Acknowledges a message that this bulk invoice was submitted. Updates its status to requested. For each line item, updates its status to submitted if it was submitting. Throws an exception if this invoice is not in requesting status.

Change in Behavior for Status Change For Unexpected Status Values

As a part of this change, there is a minor change to behavior for unexpected status transitions with all the new APIs. If the current status is not the previous status it expects, the new APIs throw an exception. The old APIs would ignore the request. For example, The `check.acknowledgeSubmission()` method throws an error if the check is not in requesting status. To avoid exceptions, message code such as asynchronous reply plugins can get the status before calling the API. For example, confirm that a check is still in requesting status before calling `check.acknowledgeSubmission()`. Contrast this with the behavior with the deprecated message-based methods if the check status was not requesting, for example if the check updated to issued or cleared already. If you use the message-based methods in such cases during message acknowledgment, ClaimCenter ignores the status change for the Ack and the check does not transition to requested.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Update financials integration code to use the new domain methods on financials objects, rather than the deprecated message-based methods.

Validation Plugin Removed

The validation plugin interface (`IValidationAdapter`) is no longer supported. Use the Validation rule set instead.

UPGRADE TASK

Priority: Before starting up the server

Summary: If you used the validation plugin, refactor to use the Validation rule set instead.

Multicurrency-related Changes

If you use multicurrency features of ClaimCenter, you must use Initial Reserves rule set instead of using the InitialReservesAdapter plugin due to multicurrency support in the rule set.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: If you use multicurrency and the initial reserves plugin, refactor to use Initial Reserves rule set instead.

If you use multicurrency mode, you must make a minor change to your policy search plugin (IPolicySearchAdapter) implementation. You must set the currency on the policy by setting the Policy.Currency property.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: If you use multicurrency and the policy search plugin, update your plugin to set the currency for a policy.

The type CurrencyAmount no longer extends BigDecimal. The CurrencyAmount type now is a plain Java object (a POJO). This change primarily affects Java code. Because Gosu language automatically coerces between CurrencyAmount and BigDecimal, the impact to all Gosu code is typically minimal. Check for any compilation errors and warnings in both Gosu and Java code, however.

In single currency mode, to get a BigDecimal value version of the amount, in Gosu get the CurrencyAmount.Amount property (in Java, call CurrencyAmount.getAmount()). It is safe to ignore the currency in single currency mode. In multiple currency mode, always take account of a money amount's currency. Thus, for multiple currency mode it is always best to convert your code to use solely CurrencyAmount values instead.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: If you use multicurrency, review your code that manipulates currency amounts and big decimal values. Refactor the code to use currency amounts to preserve the currency information.

As a consequence of the CurrencyAmount change, the rules during math operations for mixing CurrencyAmount with BigDecimal values changed. For example, you can add two CurrencyAmount values together. However, you cannot add a CurrencyAmount to a Number. Look for any compilation errors. The easy fix in single currency mode is to set a BigDecimal value version of the amount, in Gosu get the CurrencyAmount.Amount property (in Java, call CurrencyAmount.getAmount()). However, in multiple currency mode, always convert code to operate with CurrencyAmount values exclusively. That way, math operations automatically enforce that the currencies match on the money amounts you are adding, subtracting, or other mathematical operations.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: If you use multicurrency, check for compilation errors in math operations that mix currency amounts and big decimal values. Refactor the code to use currency amounts to preserve the currency information.

In the `IClaimFinancialsAPI` web service, there are new method signatures of `addClaimFinancials` and `addClaimFinancialsWithValidation` that accept an exchange rate parameter.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: If you use multicurrency, check for compilation errors in math operations that mix currency amounts and big decimal values. Refactor the code to use currency amounts to preserve the currency information.

The way that you set amounts on transaction line items changes. You cannot set the `TransactionLineItem.Amount` property. Instead, use the following code as an example of how to set the amount.

```
lineItem.setTransactionAmountAndUpdate(CurrencyAmount.getStrict(someAmount, Currency.TC_USD))
```

✓ UPGRADE TASK

Priority: Before starting the server

Summary: If you use multicurrency, check for compilation errors when setting amounts on transaction line items. Refactor the code to use the method that sets and updates the amount instead of setting the amount property.

Claim Financials Web Service API Changes

In the `IClaimFinancialsAPI` web service interface, there are new methods for voiding and stopping checks: `voidCheck` and `stopCheck`.

Messaging API Changes

The following subtopics describe changes related to messaging.

Message 'getKey' and 'addKey' Removed

The `Message` entity methods `getKey` and `addKey` were deprecated in previous releases and now are removed. Replace any remaining usages of these with the entity methods `putEntityByName()` and `getEntityByName()` respectively.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Remove any references to previously-deprecated and now-removed message methods `getKey` and `addKey`. Replace with the renamed APIs.

Message Duplicate Handling Changes

In ClaimCenter 5.0, when a message is completed (acked or skipped), the application moves the message to the message history table. This means that if there is a duplicate message, the reply plugin must look in the message history table instead of the message table. After your code finds the original message, report the duplicate on the **message history** (`MessageHistory`) entity. Since you never find a duplicate message in the regular message table, it is not possible to call `reportDuplicate()` on the `Message` entity.

Instead, the `MessageFinder` interface has new methods to allow the reply plugin to find message history entities given the original message ID or the sender reference ID. The `MessageFinder` is the object you use within your `MessageReply` plugin to find messages during asynchronous message reply handling.

The new `MessageFinder` methods are as follows:

- `findHistoryByOriginalMessageID(int originalMessageId)` - find message history entity by original message ID
- `findHistoryBySenderRefID(String senderRefID, int destinationID)` - find message history entity by sender reference ID

After you find the original message, use these methods in your reply plugin to call `reportDuplicate` on it. The `MessageHistory` entity now has a new `reportDuplicate` method so you can report duplicates from that entity.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: Review code relating to handling duplicate messages and message history. Ensure your code relies on the new behavior.

Message Finder 'Find By Sender Ref ID' Changes

On a related note, the method signature for `find` on `MessageFinder` that does not take the destination ID is now deprecated. In other words, the following method variant is now deprecated:

```
MessageFinder.findBySenderRefId(String senderRefId)
```

Instead of this method, use the method variant with the destination ID as an additional argument:

```
MessageFinder.findBySenderRefId(String senderRefId, int destinationID)
```

This version of the method is strongly preferable because the sender reference ID does **not** have to be unique across destinations. Specifying the destination ID reduces ambiguity of the request. Guidewire strongly encourages you to change to the new method signature.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: Review messaging code to update method signature for the find by sender reference ID method.

New Automated Retry on Error With Specified Date

There is a new feature to automatically retry a message at a later specified time when an error is encountered.

Message entities now have a new method signature for the `reportError` method that takes a `Date` object representing the retry time. In other words, the signature is `message.reportError(retryDate : Date)`. The messaging send queue thread on the batch server schedules a task to retry your message. This resends the message in the same way as if at that specified time you retried the message.

To do this manually, go to the Administrative interface in the messaging administration area, click a message, and then click **Retry**.

Message 'MessageSink' Property Renamed to 'DestinationID'

On the `Message` entity, the `MessageSink` column now is called `DestinationID`. This change was made to accurately reflect that this represents a messaging destination plugin ID, rather than the now-unsupported message sink system in previous releases. If you used this property in any Gosu find queries, you must replace `MessageSink` with `DestinationID` in those queries. There is no automatic upgrade for this change.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Review messaging code and any 'find' queries that reference the message property `MessageSink`. Update to instead use new name `DestinationID`.

Deprecated Properties on 'Message' and 'MessageHistory' Entities

The following columns are now deprecated in both Message and MessageHistory entities:

- OptionalString
- OptionalInt
- OptionalMoney
- AckCode
- MessageCode
- DuplicateCount (only on the Message entity, not MessageHistory). The DuplicateCount property only applies to inactive messages, which only exist in the message history table.

Other than the DuplicateCount property, the deprecated properties represent general-purpose properties as String values. Guidewire recommends you replace any usage of this properties with custom fields with more specific names. For example, instead of using OptionalString to represent a foreign key in an external system, create a data model extension property. Name your new property to indicate the specific system and ID, for example if it represents an ID for an external check printing system, perhaps call it CheckPrintingID.

IMPORTANT Guidewire strongly recommends updating your code to use custom data model extensions to replace deprecated message properties such as MessageCode and OptionalString.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Update your code to use custom data model extensions to replace deprecated message properties for message code and optional String data.

Message Context Old Get/Set Destination Methods Removed

The MessageContext object that you use in your Event Fired rules no longer has setDestination and getDestination methods. These methods were previously deprecated. Instead, use the supported MessageContext methods called setDestID and getDestID.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Check messaging code for compilation errors due to removed message context method for set and get destination and replace with renamed methods.

Message Context Create Late Bound Method Removed

The MessageContext object that you use in your Event Fired rules no longer has the createLateBound method. This method was already deprecated in previous releases.

To implement late binding, use a MessageRequest plugin implementation to populate the message payload based on the latest entity data at send time. For more information, see “Implementing a Message Request Plugin” on page 178 in the *Integration Guide*.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Check messaging code for compilation errors due to removed message context method for old-style late binding in messaging code.

Document Management Plugin Implementation Location Changes

Some files changed packages:

- All document plugin interface definitions moved to `gw.plugin.document`
- Implementations of the interfaces moved to `gw.plugin.document.impl`
- All classes used by documentation management interfaces or built-in implementation moved to `gw.document`. In other words, classes that are method return types and parameter types for documentation management interfaces or built-in implementation moved to `gw.document.*`. As a consequence, look for compilation errors related to change in packages of these objects.

✓ UPGRADE TASK

Priority: Before starting up the server

Summary: Check for compilation errors related to package changes of document management plugins and plugin implementations.

Parameterization of Types Stripped from Java External Entities

If any plugin definition uses parameterized types, ClaimCenter now strips parameterization from those types when generating external entities in the Java entity libraries. This applies to both method parameters and return types. For example, this means that if a plugin interface returns the type `List<User>`, now it simply returns the type `List`.

In nearly all cases, this change does not generate an upgrade impact since the non-parameterized type is covariant with the parameterized type. In other words, using `List` as an example, you can assign a variable of type `List<Object>` or any other `List<X>` to a variable defined as the type `List`.

In some extremely rare cases you might get a Java compile error that might require a minor change to match the defined type of a parameter or return type.

✓ UPGRADE TASK

Priority: Before starting up the server

Summary: Review Java code for compilation errors that require minor change related to stripping parameterization of types in external entities.

New Plugin Registry (Old Version Deprecated in Gosu)

ClaimCenter changed the way to get information about registered plugins from Gosu.

In ClaimCenter 5.0.x, you could use the `PluginRegistry` object with code such as the following:

```
PluginRegistry.getPlugin(com.guidewire.cc.plugin.document.IDocumentContentSource)
```

That `PluginRegistry` object is now deprecated from Gosu. In ClaimCenter 6.0, use the new class `gw.plugin.Plugins`.

To use it, call the `Plugins.get(INTERFACENAME)` static method and pass the plugin interface name as an argument. It returns a reference to the plugin implementation. The return result is properly statically typed so you can directly call methods on the result.

For example:

```
uses gw.plugin.Plugins

var plugin = Plugins.get( IBillingSummaryPlugin )
try{
    plugin.updateAccountBillingSettings( accountNumber, this )
}catch(e){
    e.printStackTrace()
```

```
        throw new DisplayableException(e.Message)
    }
```

You can also check to see if a plugin is enabled using the `isEnabled` method.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Update any Gosu deprecated references to the plugin registry class. This does not affect Java code.

For more information, see “The Plugin Registry” on page 122 in the *Integration Guide*.

IMPORTANT Although the `PluginRegistry` object is deprecated in Gosu, it is **not** deprecated in Java. From Java code, continue to use the `PluginRegistry` class.

IMessagingToolsAPI Web Service Interface Changes

The `IMessagingToolsAPI` web service interface changed in several ways:

- There is a new method `ackMessage` for acknowledging a message from an external system.
- The existing method for this purpose, `acknowledgeMessage`, is now deprecated.

The behaviors are almost the same, however the deprecated one returns the type `MessageOptionalFields`, which is a deprecated class. To reduce the upgrade impact of a type change, Guidewire included the new method with a different name.

The new version `ackMessage` returns a boolean value. It returns `true` if and only if ClaimCenter finds the message.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Update external integration code that uses the messaging tools API method for acknowledging a message, and use the new version.

IDataExtractionAPI Web Service Interface Deprecated

The web service interface for template execution, `IDataExtractionAPI` is deprecated.

Instead of relying on this web service:

- Write custom web services that address specific integration points
- Name your methods in your API appropriate for the integration point, rather than a general purpose template API. For example, if the template generates notification emails, name your method `getNotificationEmailText`.
- Design your APIs to use method arguments with types specific to each integration point.
- Do not pass template data (or anything with Gosu code) directly to ClaimCenter for execution. Instead, store template data only on the server and pass only parameters to your API.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: If any integration code used the template execution web service (the data extraction API), convert these cases to custom web services for each integration point.

IClaimAPI Web Service Interface Changes

The IClaimAPI web service interface changed in several ways:

- The method `markArchiveReady` was renamed to `scheduleForArchive`.
- There is a new version of the `getClaimInfo` method that does not use the `ObjectFilter` argument, which is a deprecated class. The new method signature with no `ObjectFilter` simply retrieves the entire `ClaimInfo` entity with no filtering. In other words, it acts the same as the deprecated method signature with the `ObjectFilter` parameter with a `null` value.
- Because of the new FNOL Mapper implementation, there are two new methods to support that tool. For details, see “FNOL Mapper is Now a Server-Side Tool” on page 94.
- In ClaimCenter 5.0, the IClaimAPI methods `completeActivity` and `skipActivity` methods set the status of the activity but did not run standard rule sets or set the `CloseDate/CloseUser` properties. Starting in ClaimCenter 6.0.0, these methods run the standard rule sets for skipped or completed activities, and now set the `CloseDate/CloseUser` properties. If you want the old behavior, add your own equivalent version in your own custom web service.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Update any integration code that uses the claim API web service to handle the archiving method name change, and other changes listed in the What's New and Changed book.

IClaimFinancialsAPI Web Service Interface Changes

The claim financials web service IClaimFinancialsAPI changed in several ways:

- additional method signatures of `addClaimFinancials` and `addClaimFinancialsWithValidation` that take exchange rates. For details, see “Multicurrency with New Financials” on page 201.
- additional method signatures of `applyForeignExchangeAdjustmentToCheck` and `applyForeignExchangeAdjustmentToPayment` that take a `BigDecimal` value for the new reporting currency amount.
- New method for stopping a check: `stopCheck`. It takes a check ID and returns nothing.
- New method for voiding a check: `voidCheck`. It takes a check ID and returns nothing.

Document Management Plugin Changes for Availability

The document management plugins now have two extra methods that you must implement to tell ClaimCenter whether the document management system is available right now for certain actions.

- `InboundAvailable` - ClaimCenter gets the `InboundAvailable` property from the plugin. If the external DMS is available for the following methods: `addDocument`, `updateDocument`, `removeDocument`. Return `true` if and only if those services are available. From Java, this appears as a method called `isInboundAvailable`.
- `OutboundAvailable` - Similarly, ClaimCenter gets the `OutboundAvailable` property from the plugin to test if the external DMS is available for the following methods: `isDocument` and `getDocumentContentsInfo`. Return `true` if and only if those services are available. From Java, this appears as a method called `isOutboundAvailable`.

This affects the following plugin interfaces:

- `IDocumentContentSource`
- `IDocumentMetadataSource`

- IDocumentMetadataSourceBase

✓ UPGRADE TASK

Priority: Before starting the server

Summary: For document management storage plugins, add new properties (for Gosu) or methods (for Java) for inbound and outbound method availability.

Geocoding Plugin Changes for Map Options

The geocoding plugin (GeocodePlugin) has a new method signature for `getMapForAddress`. The new version has an extra `String` value for implementation-specific options. You can use this to customize the application to pass extra options to your plugin as part of the request embedded in the `String` value. The format is for you to decide.

Geocoding plugin implementations must always ignore unknown options.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Updating geocoding plugin with a new method signature to get maps with options.

Testing Clock (For Development Use Only)

There is a new plugin for testing complex behavior over a long span of time, such as multiple billing cycles or timeouts that are multiple days or weeks later. The `ITestingClock` plugin is for development (non-production) use only. It programmatically changes the system time to accelerate the perceived passing of time within ClaimCenter. For details, see “Testing Clock Plugin (Only For Non-Production Servers)” on page 336 in the *Integration Guide*.

WARNING The testing clock plugin is for application development only. You must never use it on a production server.

Summary of Plugin Changes, Additions, and Removals

The following table summarizes plugin additions and removals in ClaimCenter 6.0:

Interface	New, Changed, Removed	Description
<code>IProcessesPlugin</code>	New	Instantiates new batch processes. This new plugin helps support the new feature for customer-created batch processes. For more information, see “Custom Batch Processes” on page 325 in the <i>Integration Guide</i> .
<code>IStartablePlugin</code>	New	Creates new plugins that immediately instantiate and run on server startup. For more information, see “Startable Plugins” on page 128 in the <i>Integration Guide</i> .
<code>IPreupdateHandler</code>	New	Implements your preupdate handling in plugin code rather than in the built-in rules engine. See “Preupdate Handler Plugin” on page 337 in the <i>Integration Guide</i> .
<code>IDocumentContentSource</code> <code>IDocumentMetadataSource</code> <code>IDocumentMetadataSourceBase</code>	Changed	New <code>InboundAvailable</code> and <code>OutboundAvailable</code> properties. See “Document Management Plugin Changes for Availability” on page 103.

Interface	New, Changed, Removed	Description
IMessagingToolsAPI	Changed	There is a new method <code>ackMessage</code> for acknowledging a message from an external system. The existing method for this purpose, <code>acknowledgeMessage</code> , is now deprecated. The behaviors are almost the same, but have different return types. For details, see “IMessagingToolsAPI Web Service Interface Changes” on page 102.
IEncryptionPlugin	New	Encodes or decodes a <code>String</code> based on an algorithm you provide to hide important data, such as bank account numbers or private personal data. ClaimCenter does not provide any encryption algorithm in the product. ClaimCenter simply calls this plugin implementation, which is responsible for encoding an unencrypted <code>String</code> or reversing that process. The built-in implementation of this plugin does nothing. See “Encryption Integration” on page 401 in the <i>Integration Guide</i> .
IValidationAdapter	Removed	This plugin is no longer supported. Use the Validation rule set instead.
IBackupWithholdingPlugin	New	Handles backup withholding on checks. For details, see “Deduction Calculations for Checks” on page 235 in the <i>Integration Guide</i> .
GeocodingPlugin	Changed	New support for passing options in the map request to the plugin. See “Geocoding Plugin Changes for Map Options” on page 104.
ITestingClock	New	Changes the system time programmatically to accelerate the perceived passing of time within the application. It helps test complex behavior over a long span of time, such as multiple billing cycles or timeouts that are multiple days or weeks later. WARNING: This plugin is for development (non-production) use only. For details, see “Testing Clock Plugin (Only For Non-Production Servers)” on page 336 in the <i>Integration Guide</i> .
IPolicySystemNotificationPlugin	New	New system for notifying policy systems of various issues, including large loss notification.
IPolicyRenewalAlert	Deprecated	Deprecated because the <code>IPolicySystemNotificationPlugin</code> now serves this purpose.
SegmentationAdapter	Deprecated	Deprecated in favor of segmentation rules. See “Segmentation Plugin Deprecated” on page 107 in the <i>New and Changed Guide</i> .

Plugin Template Suffix is Now ‘.gsm’

Some older plugin interfaces use text generated by these templates to pass parameters from the Guidewire application to a plugin implementation (written in either Java or Gosu). In ClaimCenter 6.0, plugin template files now have the `.gsm` suffix, not `.gs`. There is no upgrade task for you for this change. The configuration upgrade tool automatically makes this change for all your plugin templates. For more information, see “Writing Plugin Templates in Gosu” on page 120.

Web Service Changes, Additions, and Removals

The following table summarizes SOAP API additions, removals, and changes in ClaimCenter 6.0:

Interface	New or Changed	Description
IProfilerAPI	New	APIs to control the Guidewire profiling system. For more information, see “Profiling Web Services” on page 87 in the <i>Integration Guide</i> .

Interface	New or Changed	Description
PCClaimSearchIntegrationAPI	New	APIs to search and retrieve claims from Guidewire PolicyCenter or other policy systems. For more information, see “Claim Search Web Service For Policy System Integration” on page 454 in the <i>Integration Guide</i> .
IClaimAPI	Changed	Various changes, see “IClaimAPI Web Service Interface Changes” on page 103.
IClaimFinancialsAPI	Changed	Various changes, see “IClaimFinancialsAPI Web Service Interface Changes” on page 103.

Authentication Integration Changes

The plugin interface `DBAuthenticationPlugin` changed slightly. In previous releases, the one method parameter for `retrieveUsernameAndPassword` was a `Map` object. It is now the name of the database (as a `String`) for which the application requests authentication information. This will match the value of the `name` attribute on the database or archive elements in your `config.xml` file.

Perhaps using the old plugin interface you passed additional properties in the `Map`. If you need to pass additional optional properties such as properties that vary by server ID, pass parameters to the plugin in the Studio configuration of your plugin. Get these parameters in your plugin implementation using the standard `setParameters` method of `InitializablePlugin`. For more information, see “Deploying Gosu Plugins” on page 111 in the *Integration Guide*.

Enforcement of Web Service Type Name Conflicts

Within web services that you publish from ClaimCenter, it is invalid to publish two types with the same name in the set of types for method arguments and return types. This is true even if the types have different packages and even if they are in different published web services.

For example, suppose you had two web services:

- Web service `MyAPI1` exposes an entity `entity.Policy` as a return type.
- Web service `MyAPI2` exposes a Gosu type `mypackage.integration.Policy`

If you try to access locally-published SOAP types, both evaluate to `soap.local.entity.Policy`. Similarly, there is ambiguity in the WSDL and generated Java libraries.

This was invalid in previous releases of ClaimCenter, but the SOAP and Gosu parsers did not flag name conflicts. The Gosu parser now flags any name conflicts it finds as compile errors.

If Gosu identifies a conflict between two types, change the class name on one of them. This resolves the issue. If you have more questions about this, contact your Guidewire customer service representative.

To make debugging during upgrade easier, ClaimCenter 6.0.1 has a new configuration parameter called `AllowSoapWebServiceReferenceNamespaceCollisions`. If set to `true`, these error messages become warnings. Use this for development and debugging until you have time to rename your classes to fix the namespace collision. This setting is `false` by default. It is unsafe to set to `true` for production servers. If there is an edge case where you think you cannot rename classes to avoid namespace collisions, please contact Guidewire Customer Support.

Metropolitan Timeout Changes

The built-in Metropolitan reports integration has several new configuration settings for timeouts. For details, see “Customizing Metropolitan Timeouts” on page 399 in the *Integration Guide*.

Segmentation Plugin Deprecated

The segmentation plugin interface (ISegmentationAdapter) is now deprecated. Instead, use the segmentation rule set.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Remove any uses of the deprecated segmentation plugin and replace with rule set code.

New and Changed in Rules in 6.0

This topic covers what is new and changed with the ClaimCenter business rules in ClaimCenter 6.0.

This topic includes:

- “New in Rules in ClaimCenter 6.0” on page 109
- “Changes in Rules in ClaimCenter 6.0” on page 110

New in Rules in ClaimCenter 6.0

Guidewire has made the following changes to rules:

- Create Activity Pattern by Code Rather Than by ID

Create Activity Pattern by Code Rather Than by ID

In ClaimCenter 6.0, Guidewire recommends that you create activity patterns using its code value, rather than its ID value. To aid in obtaining the activity pattern through its code value, Guidewire exposes the following method:

```
entity.ActivityPattern getActivityPatternByCode( code )
```

For more information, see “Using Activity Patterns in Gosu” on page 455 in the *Configuration Guide*.

IMPORTANT Although it is possible to work with activity patterns in GScript by using the pattern ID, Guidewire recommends that you *always* use the code value instead. Guidewire does not require that an activity pattern code be unique at this time. However, in general practice, Guidewire recommends that you do so as Guidewire intends to make this a requirement in a future release.

Changes in Rules in ClaimCenter 6.0

This topic describes what has changed with rules in ClaimCenter 6.0. These include:

- “ClaimCenter Stores Rules as Gosu Classes” on page 110
- “Rule Set Category Name Changes” on page 111
- “Removed Rules” on page 111
- “Rule Debugger Menu Changes” on page 112
- “Resource Export and Import Removed” on page 112

ClaimCenter Stores Rules as Gosu Classes

Prior to ClaimCenter 6.0, ClaimCenter stored business rules as XML files. With the release of ClaimCenter 6.0, Guidewire stores business rules as Gosu classes. However, you still access and edit these files through Studio Resources → Rules.

IMPORTANT As part of the upgrade utility, ClaimCenter provides an automatic upgrade step to convert your existing XML-based rules to Gosu-based rules. See the *Upgrade Guide* for details.

Rule Sets

During upgrade conversion, the upgrade utility converts an existing rule set into the following:

- a rule set Gosu class
- a directory that contains all of the child rules for that rule set, which it names using the convention `ruleset_dir`

The rule set Gosu class contains the following upgraded rule set info:

- name (in the `RuleSetName` annotation)
- description
- type name (the Gosu class)

The directory contains the following:

- an `order.txt` file (representing the rules under this rule set)
- other rules for this class

With the release of ClaimCenter 6.0, Guidewire defines the order of the top level rules (formerly in `ruleset.xml`) in file `order.txt`, under the rule set directory. File `order.txt` contains relative type names for the rules to run.

Rules

Individual rules follow the same directory structure as rule sets. There is a primary Gosu class, a subdirectory (name of rule + `_dir`) and an `order.txt` file.

The rule set Gosu class contains the following:

- rule set name (in the `RuleName` annotation)
- condition (converted into the `doCondition` method)
- action (converted into the `doAction` method)

The `messageContext` property used in the condition and action is converted as arguments to the `doCondition` and `doAction` methods. Furthermore, the actions are converted into the `actions` parameter of the `doAction` method.

The action and condition are wrapped in the special tokens `/*start00rule/` and `/*end00rule*/`. These must be present or Studio cannot parse the file.

With the release of ClaimCenter 6.0, Guidewire defines the order of subrules in `file order.txt`, along with the relative type names for the rules. Subrules under a rule follow the same process, recursively.

Rule Set Category Name Changes

With the release of ClaimCenter 6.0, Studio rule set names reflect their underlying package names. These package names encode the rule set type. The Preupdate and Validation rules encode the name of the root object in the package name.

Thus, you see a slightly different version of these names in the **Resources** hierarchy tree. For example, the former *Post-Setup* rule category becomes *Postsetup*. The former *Transaction Set Post-Setup* rule set becomes *TransactionSetPostsetup*.

Rule Set Category Creation through Studio

With the release of ClaimCenter 6.0, it is possible to create new rule sets and rule set categories through Guidewire Studio. In previous releases, you could only create new rules. To create a new rule set or rule set category, select one of the following:

- The **Rule Sets** node (to create a new rule set category)
- A rule set category (to create a new rule set)

Use the right-click menu **New** functionality to create the rule set category or rule set. For more information, see “Working with Rules” on page 167 in the *Configuration Guide*.

Modified setApprovingUser Method

Guidewire has modified the `entity.setApprovingUser(user, group)` method to verify that the user is not inactive or retired. If either of these conditions exist (or either the user or group value is `null`), the method returns `false`. Otherwise, it returns `true`. *Only call this method in the context of the Approval Routing rule set.*

Removed Rules

Guidewire has removed the following sample rules from the ClaimCenter base configuration due to references to no longer used entity properties.

For the Claim entity, the removed properties are:

- `DeductibleStatus`

For the Transaction entity, the removed properties are:

- `NetAmount`
- `OriginalAmount`
- `DeductibleAmount`
- `DeductibleSubtracted`
- `DeductiblePaid`

Claim Exception Rule Set - Removed Rules

- `CEWC0010 - Open - trigger deductible recovery activity`
- `CEWC0020 - Closed - trigger deductible recovery activity`

Claim Pre-Setup Rule Set - Removed Rules

- `CPS00010 - Set Deductible Status to Unpaid`
- `CEWC0020 - Closed - trigger deductible recovery activity`

Transaction Post-Setup Rule Set - Removed Rules

- TPOWC010 - Set Deductible status to Paid - WC

Rule Debugger Menu Changes

Guidewire has modified the rule debugger by removing the following command from the Studio **Debug** menu:

- **Toggle Rule Breakpoint**

The rule debugger fully supports setting breakpoints on rule condition statements.

Resource Export and Import Removed

Prior to ClaimCenter 6.0, Guidewire provided several different Studio tools to export rules and re-import them. With the release of ClaimCenter 6.0, Guidewire has removed these tools, primarily as a consequence of moving the business rules from XML to Gosu code. In particular, Guidewire has removed the following:

- The ability to set a resource context by selecting a rule, then selecting **Resource Context** from the right-click menu.
- The ability to create a report file (**Tools** → **Rule Repository Report**) that contains an HTML version of the ClaimCenter business rules.
- Export resource types into a single XML file by selecting **Export** from the right-click **File** menu.
- Import an (exported) XML resource file into Studio file system by selecting **Import** from the right-click **File** menu, then choosing an XML file to import.

New and Changed in Reporting in 6.0

This topic covers what is new and changed with Standard Reporting in ClaimCenter 6.0.

This topic includes:

- “New in Reporting in ClaimCenter 6.0” on page 113
- “Changes in Reporting in ClaimCenter 6.0” on page 114

New in Reporting in ClaimCenter 6.0

This topic describes what is new in Standard Reporting in ClaimCenter 6.0. These include:

- New Claim Health Metric reports. See “Claim Health Metrics Reports” on page 92 in the *Reporting Guide*.
- New Sree Properties
- New Build Properties

New Sree Properties

Guidewire has added the following properties to the `sree.properties` file:

- `repository.audit.enabled`
- `schedule.auto.start`

For more information, see “Troubleshooting Guidewire Standard Reporting” on page 111 in the *Reporting Guide*.

New Build Properties

Guidewire has added the following property to the `build.properties` file:

- `build.sree_home`

For more information, see “Set Up the Reporting Configuration Files” on page 25 in the *Reporting Guide*.

Changes in Reporting in ClaimCenter 6.0

This topic describes what has changed with reporting in ClaimCenter 6.0. These include:

- Reporting Server Version Updates to Version 10.1
- Configuring Reports to Access Claim Information
- Guidewire Standard Reporting Tables Modifications
- Changes to InetSoft Charts

Reporting Server Version Updates to Version 10.1

With the release of Guidewire ClaimCenter 6.0, Guidewire uses InetSoft Style Report Enterprise Edition, version 10.1 as the reporting server. For a complete list of ClaimCenter 6.0 compatible software, see “Installation Environments Overview” on page 12 in the *Installation Guide*.

If you created custom reports, then it is possible that you need to modify your existing reports. See the following for details:

- “Upgrading from ClaimCenter 4.x to ClaimCenter 6.x” on page 43 in the *Reporting Guide*
- “Upgrading from ClaimCenter 5.x to ClaimCenter 6.x” on page 59 in the *Reporting Guide*

Configuring Reports to Access Claim Information

With the release of ClaimCenter 6.0, Guidewire has modified the process of configuring ClaimCenter reports to access claim information (through drill-down reports). In releases prior to ClaimCenter 6.0, you used a `beanLink` property to set the report server URL, used in the report template `.srt` files. With the release of ClaimCenter 6.0, you use an `APPURL` property in `sree.properties` to set this value.

Guidewire provides updated report templates for all base configuration reports. However, if you create custom reports, then you need to take this into account. See “Configuring Reports to Access Claim Information” on page 107 in the *Reporting Guide* for more information.

Guidewire Standard Reporting Tables Modifications

In ClaimCenter 6.0, Guidewire has added the following reporting tables:

- `cc_incident`
- `cc_ActivityPattern`
- `cc_subrogationsummary`
- `cc_subroadverseparty`
- `cctl_FaultRating`
- `cctl_LossPartyType`
- `cctl_subrogovernmentinvolved`
- `cctl_subroschedrecoverytype`
- `cctl_subroclosedoutcome`
- `cctl_languagetype`

In ClaimCenter 6.0, Guidewire has removed the following reporting tables:

- `cctl_state`
- `cctl_detailedbodyparttype`

Changes to InetSoft Charts

InetSoft has changed the engine that generates charts in the InetSoft version 10.1 release. InetSoft still generates charts that were created prior to this change. However, you can consider these charts as deprecated. At some future time, InetSoft will no longer support these types of charts. Guidewire strongly encourages you to recreate these charts as part of your InetSoft upgrade process.

Note: For information on InetSoft charts and how to work with them, see the *InetSoft Technology Report Designer Guide, version 10.1*

UPGRADE TASK

Priority: Before your next upgrade

Summary: Recreate InetSoft charts using InetSoft Report Designer 10.1.

part II

What's New and Changed in 5.0

New and Changed in ClaimCenter 5.0

New in ClaimCenter 5.0

This topic describes new features in ClaimCenter 5.0. These include:

- Address Completion
- Assessments
- Catastrophes
- Claim Archiving
- Claim Associations
- Injury Incidents
- Lines of Business
- Multiple Currencies
- Question Sets
- QuickJump Box
- Regional Holidays
- Security on Notes
- Service Provider Management
- Subrogation
- User Interface
- Workflows
- Zone Mapping and Autofill

Address Completion

Autofill is a feature of ClaimCenter that helps you complete addresses during data entry. There are several important ways that autofill can help complete an address:

- When autofill receives a city and state, it can return the county and ZIP code, or return a list of ZIP codes.
- When it receives a ZIP code, it can return the city, county and state, or lists when there are more than one.
- When it receives a combination of address elements that do not constitute a valid address, autofill can block its submission and display a warning message. Address elements include things data such as city, state and ZIP code. The user may override this warning and add the address to the database by simply re-submitting the address, or may edit the address.
- Whenever a list is returned, the user can directly select the correct value from the list.
- For countries other than the USA, autofill now uses the internationalization feature of the zone mapper to provide similar autofill capabilities for other kinds of address elements (zones).

Autofill can occur without the user moving the cursor to or away from autofilled fields, and does not require any other user action.

For more information, see “Understanding Autofill and Zone Mapping” on page 61 in the *Contact Management Guide*.

Assessments

Assessment is the process of evaluating the value of lost or damaged property, then providing and monitoring the services required to indemnify the insured and cover related expenses.

ClaimCenter incorporates the Assessments feature into both auto and property claims. This solution includes:

- Maintaining lists of sources (evaluators, or assessors).
- Itemizing, then categorizing (grouping) property for assessment.
- Managing documents and notes associated with the assessment process.
- Sending work orders to multiple sources to perform evaluations.
- Collecting and evaluating the estimates and quotes generated by the work orders.
- Agreeing to the loss value (this is typically a negotiation between the claimant an adjuster based on the assessments obtained).
- Providing the necessary services to indemnify the insured for the loss - either repair or replacement.
- Evaluating the quality of the indemnification.
- Maintaining a status display of the assessment work orders and repair orders.

For vehicle losses, providing timely assessment services is a key component of controlling leakage. Ideally, the end of every first notice of loss (FNOL) conversation concerning an auto loss includes the insured being told:

- Where and when to have the damaged vehicle assessed.
- The name of the appraiser.

Otherwise, the insured will be happy to drive the replacement rental car forever.

ClaimCenter provides one assessment process for each vehicle, building, and group of property items. It is available during the New Claim wizard, as well as by accessing the claim’s Loss Details page later. The assessments feature is an extension to incidents, hence exposures.

For more information, see “Assessments” on page 109 in the *Application Guide*

Catastrophes

The term *catastrophe* refers to a single or series of closely related incidents that cause a significant amount of losses. Carriers often group claims by the catastrophes that caused them. This helps the carrier:

- Estimate the severity of the catastrophe itself, and its potential liability due to the catastrophe.
- Estimate the reserves it must set aside to cover future claims from the catastrophe.
- Manage its resources (such as mobile adjusters) in responding to the catastrophe.
- Create reports about the catastrophe and its financial consequences for the carrier.

ClaimCenter maintains a list of catastrophes which affect the carrier's business, and can associate one catastrophe from this list with a claim.

ClaimCenter defines a catastrophe by these three characteristics:

- A *date range* (a start and end date).
- A geographic *region* (a list of US states; this has not been internationalized).
- One or more *perils* (a peril is a combination of a Loss Type - such as Property - and Loss Cause - like wind).

ClaimCenter assists carriers handle catastrophes in two ways:

- It defines and maintains a list of catastrophes. See "The Catastrophe List" on page 116 in the *Application Guide*
- It associates at most one catastrophe to a claim. See "Associating Catastrophes and Claims" on page 118 in the *Application Guide*.

For more information, see "Catastrophes" on page 115 in the *Application Guide*.

Claim Archiving

Archiving is the process of moving claims from the primary ClaimCenter database into a secondary database. You may still search for, retrieve and work with archived claims, but while archived, they are a much smaller burden to the primary database.

The main advantage of archiving claims is to improve ClaimCenter performance. As it handles more claims, its database storage requirements increase, its table lengths increase, and its performance degrades. Archiving can improve performance in the face of unbounded claims growth.

The main characteristic of ClaimCenter claim archiving is its transparency. As a user, you may not even be aware of the archiving process:

- ClaimCenter itself archives claims that it determines to be archivable.
- It uses two background (batch) processes: *markforarchive*, which flags eligible claims for archiving and *archive*, which performs the archiving on flagged claims.
- You may search for claims without knowing or caring whether they are archived or not.
- Your search results contain the same claim summary information, whether the claim is active or archived.
- When you wish to work with an archived claim found by your search, ClaimCenter restores it for you.
- A restored claim is identical to a claim that has never been archived (but its History notes that it was archived and restored).

See "Archiving" on page 87 in the *Application Guide*.

New Configuration Parameters

Guidewire has also added several new configuration parameter related to claim archiving. These include the following:

- *ArchiveEnabled*, which specifies that archiving is enabled for this system

- `DbmsCounterThresholdArchiveSecs`, which sets the batch archive threshold in seconds for the archive batch process.

See “Application Configuration Parameters” on page 35 in the *Configuration Guide* for more information.

Claim Associations

Claims are not always completely independent; one claim may be related to others, and it is often useful to associate such claims with one another. For example:

- Many claims may result from the same root cause
- Claims may have the same person as the insured and the claimant:
- Multiple claims from the same claimant could represent fraud.
- The same incident may result in multiple claims.
- The same incident may result in both a parent and child claims.
- Litigation may involve related claims.

The **Associations** tab of the claim’s **Loss Details** section allows you to associate one claim with others that already exists in ClaimCenter. It also displays a table of all claims associated with each other. For details, see “Claim Associations” on page 31 in the *Application Guide*.

Injury Incidents

Injury incidents consolidate some injury information that was formerly part of workers’ compensation, PIP and auto injury claims and exposures into injury incidents, another incident subtype. For more information, see “Incidents” on page 235 in the *Application Guide*.

Lines of Business

The six typelists that define LOBs have been rewritten to be more internally consistent and free of duplications. For more information, see “Configuring Your Lines of Business” on page 307 in the *Application Guide*. In addition, Studio contains an LOB editor to enable you to define LOBs using these typelists.

Multiple Currencies

The ClaimCenter multicurrency feature allows you to make payments, write checks and collect recoveries in more than one currency. It knows the base ClaimCenter currency, but accounts for these transactions in the currencies in which they were made. Finally, it adjusts for exchange rate fluctuations that change payment (but not recovery) amounts.

ClaimCenter financial transactions have always been conducted in a single (base) currency. Using more than one currency in ClaimCenter enables you to:

- Write checks, payments, reserves, recoveries and recovery reserves in any other (transaction) currency.
- Track these transactions as a part of the usual ClaimCenter financials summary screens and calculations.
- Integrate these transactions into ClaimCenter financial totals and summaries.

Every instance of ClaimCenter continues to use a single base currency, but many tasks can now be performed using another transaction currency. In particular, a ClaimCenter user may now:

- Create payments, issue checks, and receive recoveries in any currency.
- Create reserves and recovery reserves in any currency. All reserves, no matter their currency, still contribute to the aggregate reserves and recovery reserves. ClaimCenter displays aggregate and recovery reserves in units of the base currency.
- Search for checks and recoveries in any particular currency - or in all currencies.

- Create bulk invoice payments and write their associated checks in any currency.
- View financial summaries which include all transactions, independent of their currency.
- Adjust the base currency amount of a check and its payments. You can adjust the base currency amount after the check has been cashed and the actual amount of the check in the base currency is known. This is available in the event the amount has changed because of exchange rate fluctuations.
- Add a Preferred Currency field to contacts, which tells ClaimCenter the default currency to use when writing a check to that contact (vendor).
- Enter exchange rates manually when needed, or automatically get them from an outside provider. An exchangerate batch process is provided that uses an ExchangeRateSetPlugin in order to pull exchange rate data into ClaimCenter on a scheduled basis.
- Periodically update exchange rates obtained from an outside provider.

For a complete description, see “Multiple Currencies” on page 171 in the *Application Guide*.

Question Sets

Question sets are pre-defined sets of questions, used to help an interviewer to obtain complete information; they regularize the information gathering, and creates a searchable record of the answers. A full description of them, along with how to create your own is in “Question Sets” on page 263 in the *Application Guide*. ClaimCenter implements question sets for service provider reviews and for fraud detection. See “Using Service Provider Performance Reviews” on page 123 in the *Application Guide* and “Question Sets also Evaluate Risk Potential” on page 105 in the *Application Guide* respectively. Question sets can be imported and exported. See “Importing and Exporting Administrative Data” on page 115 in the *System Administration Guide*.

The Studio editor for question sets is not available for ClaimCenter.

QuickJump Box

The **Quick Jump Box** is a fast way for you to navigate elsewhere in ClaimCenter or search for information in specific categories. Typing the name of a command and selecting **Enter** allows you jump to that location in the application. While Guidewire provides you with a number of already defined commands, the **Quick Jump Box** can be customized to define often used commands, search parameters, and permissions in Guidewire Studio. Note that for the **Quick Jump Box** to appear on the interface, there must be at least one configured command.

Regional Holidays

You may now define holidays in the user interface and make changes to these definitions without restarting the server. In addition, you may give each holiday a type (like Federal or State) and define the region in which it is valid. You specify the Region definition using the same zones in the `state.xml` typelist used for address completion. You can substitute another set of zones (by using another typelist) to define regions outside of the United States. For more information, see “Regional Holidays” on page 123 in the *Application Guide*.

Security on Notes

ClaimCenter provides a set of system permissions to provide security for all notes; you may also mark a note as ‘confidential’. You may also use these permissions to define different security types for notes and assign permissions to users that relate to these ACLs. “Notes Security” on page 252 in the *Application Guide* and “Access Control for Documents and Notes” on page 388 in the *Application Guide* describes this in detail.

Service Provider Management

An important part of claim handling is providing services to help resolve losses. This includes using or recommending a service provider such as a body shop, assessor, attorney, or health clinic. ClaimCenter provides a

mechanism to manage your carrier's service providers by gathering rating information on them. This feature enables you to select favorable providers, control claim costs, increase customer satisfaction, and increase claim processing efficiency. See "Using Service Provider Performance Reviews" on page 123 in the *Application Guide* for details.

Subrogation

Subrogation is the legal technique in which one party represents another party, using their rights and remedies against a third party. A carrier sometimes settles a claim, knowing that another party may be liable for costs. The carrier then attempts to recover those costs from the other party in behalf of their insured. The fine print of most insurance policies cedes the insured's recovery rights to their carriers.

Subrogation is often used to recover costs from the liable party's insurance company. This is usually accomplished by informal negotiations between the two carriers involved. When the third party has no insurance, however, subrogation often involves legal action and/or collection agencies.

ClaimCenter now contains a full subrogation feature. This feature is described in "Subrogation" on page 273 in the *Application Guide*.

User Interface

The User Interface is much changed and improved in this release.

Look and Feel

The layout of all screens now uses the "Ocean" styles of blue shades; it is easier to view for long time periods. In addition, the font is now Verbená, which is known for its readability, and you may change text font sizes with your browser settings. The areas that you click are larger, and when you refresh a screen, you return to the same scroll position. Extensive use is made of AJAX, so there are more fast screen updates. Dropdown menus are now properly sized and positioned; multicurrency displays include two currencies, and a circling 'in progress' icon appears when ClaimCenter is taking time to complete its actions.

Autocompletion

Autocompletion has been expanded to include not only addresses, but contact names as well. For addresses, entering a zip code selects the state and city or gives you a few choices to select from. This also works in reverse. ClaimCenter tries to autocomplete the zip code when a user specifies a city.

Info Bar

A one-line information bar appears at the top of each claim screen. This bar contains an icon depicting the policy type, policy number, name of the insured, loss date, claim status and adjuster's name and group.

The Actions Menu

A large button, labelled **Actions** >, always appears at the upper left, that opens a set of cascading menu actions that are relevant for the screen you are on. This lets you navigate quickly to related screens.

The Unsaved Work Button

A button at the top center of the User Interface gives you the chance to save your work. ClaimCenter saves all your work when you go to another screen without completing work in the current screen. The **Unsaved Work** button takes you back to all screens to allow you to finish your work. If you instead attempt to log out of ClaimCenter, you will be alerted to the fact that you have unsaved work that logging out will erase. This button is now your last chance to save your work before logging out.

Navigation Menu

Navigation bars at the left side of screens now strongly highlight the current page to remind you where you are.

QuickJump Box

The **Quick Jump Box** is a fast way for you to navigate elsewhere in ClaimCenter or search for information in specific categories. Typing the name of a command and selecting **Enter** allows you jump to that location in the application. While Guidewire provides you with a number of already defined commands, the **Quick Jump Box** can be customized to define often used commands, search parameters, and permissions in Guidewire Studio. This feature knows about your permissions and blocks unpermitted jumps. Note that for the **Quick Jump Box** to appear on the interface, there must be at least one configured command. “The QuickJump Box” on page 53 in the *Application Guide* has details.

Workflows

A workflow is a multi-step process that manages a complex business practice that rules cannot by themselves define. ClaimCenter uses a single workflow process - to implement interactions with Metro reports. You may either modify this existing workflow or generate a new one to model other complicated business processes. “Guidewire Workflow” on page 423 in the *Configuration Guide* describes this process and gives a reference to the new workflow editor, part of Studio, that simplifies creating them.

Zone Mapping and Autofill

In the process of assignment by location, ClaimCenter maps a *zone* of a location’s address with the same kind of zone in a user’s *region*. Zone mapping looks for such matches, so that a nearby user can be assigned to work on the claim or exposure. For example, if the mapping is set up to use cities as zones, zone mapping would attempt to match the accident address’ city with a city in the user’s region. If a match is found, Assignment by Location could assign that user to the claim.

Sometimes, an address may be missing the zone needed for this mapping. In this case, zone mapping can first try to complete the address by examining another zone type in the address. For example, if all cities are linked to zip codes, when an address lacks a city, the zip code could be used to find it.

In this release, zone mapping has been internationalized and made much more flexible. See “Proximity Search Using Geocoding” on page 89 in the *Contact Management Guide* for details.

This ability of zone mapping to complete addresses can also be useful for autofill. After entering a zip code, zone mapping can link it to the zip code’s city and fill in that address element. See “Proximity Search Using Geocoding” on page 89 in the *Contact Management Guide*.

Changes in ClaimCenter 5.0

This topic describes what has changed in ClaimCenter 5.0. This includes:

- Significant Assignment Changes
- Calendar Enhancements
- Configuration Parameters
- Coverage Model Changes
- Coverage Verification
- Data and Security Dictionaries
- Database Consistency Checks
- Eclipse No Longer Supported

- Email Enhancements
- Export
- Financial Changes
- Geocoding
- Localization
- Matter and Negotiation Enhancements
- Module-Based Configuration
- New Claim Wizard
- New Work Queue Parameters
- Parties Involved Enhancements
- PCF Definition
- PCF Input Widgets
- Policy Model Changes
- SIU Questionnaire
- Workflow Flexibility

Significant Assignment Changes

As of the release of ClaimCenter 5.0, Guidewire has significantly changed the way ClaimCenter performs assignment. All of the changes are backwards-compatible (except for one, described below). If a method signature has been changed, the old signature has been deprecated, rather than removed. See “Assignment in ClaimCenter” on page 97 in the *Rules Guide* for more information.

The following summary briefly describes the more important assignment changes:

- Ability to Call Assignment Methods from any GScript code.
- Assignment Method Signatures Take “Group” Parameter
- Round-Robin Assignment Performed Over Arbitrary Set of User and Groups
- User-definable “Dynamic” Assignment
- Ability to Make Extension Entities Assignable
- Assignment Entity Removed

Ability to Call Assignment Methods from any GScript code.

It is now permissible to call the assignment methods from any GScript code. Previously, you could only legally call the assignment methods from within the context of a assignment rule run by the Assignment Engine.

Assignment Method Signatures Take “Group” Parameter

As part of assignment transformation, Guidewire has changed the signatures of many of the assignment rules to include a “Group” parameter. Previously—when assignment methods were only invoked from within the Assignment Engine—the group within which the assignment was carried out was implicit, as a function of the Assignment Engine's current state. The new method signatures all take an explicit Group parameter instead, which makes the methods usable from outside the Assignment Engine.

For backwards compatibility, Guidewire has added a GScript method within the assignment rules to retrieve the current Group:

```
AssignmentEngineUtil.getCurrentGroupFromES()
```

IMPORTANT Guidewire has deprecated assignment method signatures that do not have a Group parameter. Studio indicates this status by marking through the method signature in GScript code. In addition, Studio code completion does not display deprecated methods. You can, however, see them in listed in the full list of assignment methods (with Studio again indicating they are deprecated). Do NOT use a deprecated assignment method—one without the Group parameter—outside of the Assignment rules. In general, Guidewire recommends that you do not use deprecated methods at all. If your existing GScript code contains deprecated methods, Guidewire recommends that you rework your code to use non-deprecated methods.

Round-Robin Assignment Performed Over Arbitrary Set of User and Groups

Prior to this release of ClaimCenter, you could only perform round-robin based assignment within a single Group. As a consequence, several of the assignment methods did not work as expected. In this release, assignment methods that rotate over an arbitrary set of users (by using a `UserSearchCriteria` entity, for example) correctly perform round-robin assignment. ClaimCenter uses round-robin assignment among all matching users, not just those in the current Group.

User-definable “Dynamic” Assignment

Guidewire has added two new methods:

```
assignUserDynamically()
assignGroupDynamically()
```

These methods take a class that implements one of the following interfaces:

```
DynamicUserAssignmentStrategy
DynamicGroupAssignmentStrategy
```

You can now write a GScript class implementing one of these interfaces and use that class to perform assignments that meet your business needs. One primary use for this is to perform “load-balancing” assignment, in which entities are assigned to the User with the smallest number of existing assignments.

Ability to Make Extension Entities Assignable

Guidewire now permits you to make your extension entities assignable.

Assignment Entity Removed

Guidewire has removed the `Assignment` entity. In its place, Guidewire provides an `Assignable` entity. This change specifically affects any code that had been executing on events tied to the `Assignment` entity. For example, if you have existing messaging code that captures `Assignment` Events with a context object of `Assignment`, the code will no longer be valid. You need to ensure that your GScript code does not rely on the `Assignment` entity. (This includes, but is not limited to, messaging code that relies on `Add`, `Changed`, `Removed` events on that entity.)

In its place, use the new assignment data model and APIs to detect changes to assignments directly on assignable entities. You can detect assignment events directly on assignable entities. For example, the `Event Fired` rules can listen for `AssignmentAdded`, `AssignmentChanged`, and `AssignmentRemoved` events on assignable entities such as activities. In the GScript code, the root object for the message will be the `Assignable` entity (not the `Assignment` entity).

Calendar Enhancements

The legal calendar now has many different displays, and can show upcoming activities related to claims as well as matters. Displays include:

- Calendars for the current week and month, or any other start date (weekly calendars always start with the current date, the monthly calendar always starts on the previous Monday).
- All activities related to all claims and matters, all those unrelated to legal matters, or those related to matters.
- If looking at matter-related activities, either a display of all such activities or just all trial dates.
- Activities assigned any priority, or just activities of a specific priority, such as **Urgent**.

For more information, see “Activity Calendars” on page 228 in the *Application Guide*

Configuration Parameters

The `DaysClosedBeforeArchive` configuration parameter, which specifies the number of days after a claim is closed before being considered for archiving, is now set to 30 days.

Coverage Model Changes

The ClaimCenter coverage model, or how it defines Lines of Business, has been reworked. The six existing type-lists are more complete and flexible. See “Configuring Your Lines of Business” on page 307 in the *Application Guide* for a complete description. In addition, a new LOB editor, described in the Studio Guide, allow you to configure new LOBs in a simple way.

Coverage Verification

ClaimCenter uses the policy of the claim and its coverages to help you create exposures that make sense, and warn you or prevent you from creating exposures that do not. Whenever you create a new exposure, ClaimCenter looks for inconsistencies between a policy’s coverages and the loss party, loss cause, other existing exposures, and claimant’s liability.

Three table associate loss causes with appropriate exposures, loss party with appropriate exposures, and exposures on a claim incompatible with other existing exposures. See “Coverage Verification” on page 75 in the *Application Guide* for details. In addition, you may edit and extend these tables. See “Coverage Verification Reference Tables” on page 413 in the *Application Guide*.

Data and Security Dictionaries

You now generate these dictionaries with the `gwcc regen-dictionary` command. They are created in your local `ClaimCenter/dictionary` directory. Neither dictionary is now included in the application server (`.war` or `.ear`) deployment, and therefore are no longer provided as a web page served by the application server.

To view the data dictionary, open `ClaimCenter/dictionary/data/index.htm`. `ClaimCenter/dictionary/security/index.htm` takes you to the security dictionary. This means you have access to these dictionaries without having to start a server.

Database Consistency Checks

The system tool `-checkdbconsistency` can now be run for either the entire database or for a subset of tables. You can also specify to run all consistency checks or a subset of checks. For details and instructions, see “system_tools Command” on page 173 in the *System Administration Guide*.

Eclipse No Longer Supported

The only reason to use Eclipse in connection with Guidewire applications is for Java coding for integration purposes. However, it is important to know that a crucial aspect of Guidewire configuration resources - namely their predominance - cannot be viewed in Eclipse.

Email Enhancements

ClaimCenter provides enhanced ability to send emails from within ClaimCenter. You may generate emails using GScript, GScript classes, and rules. GScript supports:

- Sending to different types of recipients: To:, CC:, and BCC: recipients.
- Attaching documents stored in the configured document management system.
- Using templates to populate the Subject and Body of the email.
- Defining activity patterns with an associated email template.
- Creating Activities with a Send Email button that use a particular email template.

For more information, see “Outbound Email” on page 231 in the *Application Guide*.

There is also a user interface that allows users to create emails in the course of their work.

Export

The `export_tools` command line utility for exporting administrative data has been removed. The SOAP protocol used by `export_tools` could not handle the large amounts of data being exported. You can export the same administrative data from ClaimCenter.

Financial Changes

There have been a number of changes to the way ClaimCenter handles its financials:

- You can search bulk invoices. See “Viewing Bulk Invoices From the Desktop” on page 201 in the *Application Guide*.
- Negative recoveries are allowed. See “Recoveries and Recovery Reserves” on page 149 in the *Application Guide*.
- Checks (normal and manual) may be rejected downstream. See “Downstream Denials of Recoveries and Checks” on page 160 in the *Application Guide*.
- There are no longer restraints on supplemental checks. See “Partial, Final, and Supplemental Payment Types” on page 140 in the *Application Guide*.

Geocoding

Geocoding is a method of determining the exact position of an address on the earth’s surface. Geocoding turns an address into a latitude and longitude. This information can then be used to determine the closeness, or the proximity, of two addresses. Geocoding is now integrated into ContactCenter and ClaimCenter. Its features have not changed, but have been recoded for efficiency and speed. You can find a description of Geocoding in “Proximity Search Using Geocoding” on page 89 in the *Contact Management Guide*.

Localization

The following are changes in the way you define the localization configuration:

- To localize display properties and typelists, export the English values from Guidewire Studio, translate them, and then import them back into Studio.

- Locales are now defined in the `localization.xml` file, which you can edit from Studio. The `localization_ext.xml` file no longer exists.
- The localized display properties are now imported directly into and managed by Guidewire Studio. Therefore, the `<DisplayProperties>` element in `localization.xml` is no longer needed or supported.
- The `<DisplaySettings>` element in `localization.xml` is no longer supported.

For more information, see “Localizing Guidewire ClaimCenter” on page 463 in the *Configuration Guide*.

Matter and Negotiation Enhancements

New screens reached from the **Matters** menu item on all claim screens takes you to pages where you can enter information related to legal matters and negotiations. In addition, there is a powerful new page that tracks potential and actual costs of the matter; the New Check Wizard also requires the matter type for legal cost categories. See “Legal Matters” on page 241 in the *Application Guide* for a description of these improvements and changes.

Module-Based Configuration

In previous versions of ClaimCenter, configuration resources have been arranged one way in the configuration environment and a different way in the deployed environment. Now the directory structures are identical in configuration and deployed environments - greatly simplifying the process of working with these files.

The `/configuration` directory is the only place for user-edited resources. When the application starts, a checksum process verifies that no files have been changed in any directory except for those in the `/configuration` directory. If this process detects an invalid checksum, the application will not start. In this case, you will need to overwrite any changes to all modules except for the `/configuration` directory and start again.

Never attempt to edit any file directly, even in the `/configuration` directory. Instead, use Studio to select the file to edit from the Resources tree. For many resources, Studio has built-in editors. For other files, you may specify the default editor of your choice to use for these file types. In all cases, Studio will open the resource from the module with highest precedence, and then save it in the configuration module, leaving the original file alone.

New Claim Wizard

The New Claim Wizard has been rewritten completely. Significant changes include:

- fewer required pages
- optional pages
- improved navigation
- improved work saving during data entry
- organization based on incidents
- improved layout
- new icons
- flows that change depending on the type of claim.

See “Claim Creation” on page 59 in the *Application Guide*.

New Work Queue Parameters

New parameters are available for workers processing a work queue. These parameters enable you to configure the following:

- How long a worker must sleep after exhausting all work items, ignoring notifications, before waking up and polling the database again for more work items. This can reduce unnecessary load on the database for systems with a large number of work items and workers.

- How often workers wake up automatically, even if no notification is received. This can be fine tuned to reduce excessive queries on the database.
- How many work items the worker will attempt to check out when searching for more work items.

Parties Involved Enhancements

ClaimCenter has expanded the definitions of Users and Contacts. See “Parties Involved” on page 34 in the *Application Guide*. One key feature is the ability to separate contacts into subtypes and apply different levels of security to these subtypes. See “Users” on page 370 in the *Application Guide*, and the *Guidewire Contact Management Guide* for configuring the ACLs of contact subtypes.

PCF Definition

The method for defining PCF files has changed. You no longer edit PCF XML files directly. Instead, use Guidewire Studio. For more information, see “PCF Editor” on page 181.

PCF Input Widgets

Editable input widgets in PCF used to contain an attribute called `setter` that allowed arbitrary code to set the value of the field. The `setter` attribute is now deprecated.

The `setter` attribute was often used to access items in lists and fields in maps. GScript now contains a feature to replace this type of usage. PCF editable input widgets can now use value expressions that use bracket syntax to indicate a list or map value. For example, you can now use the field value `myEntity.MyProperty[value]`. This path expression will be used for both the getting and setting of the widget.

Policy Model Changes

The ClaimCenter policy data model has been changed to make it more compatible with PolicyCenter. See “Policies and the Data Model” on page 80 in the *Application Guide* and “Policy Data Model Changes” on page 249 in the *Upgrade Guide*.

SIU Questionnaire

The SIU (Fraud) questionnaire has been rewritten to take into account the new, general method of creating question sets; it now includes both a question set and automatic SIU scoring. See “Fraud Detection Overview” on page 102 in the *Application Guide* for a description of the new question set. See “Question Sets” on page 263 in the *Application Guide* for information on how to edit this question set or create your own.

Workflow Flexibility

Workflows support complex business processes that run during specific time periods. Now you can extend that functionality through the following:

- **Define new workflow types and start them in Rules using GScript.** First, create the workflow instance based on the workflow type. Then, pass in the relevant context entities. Finally, start the workflow instance. Use the Studio Workflow Editor.
- **Create new workflow definitions** by specifying one or more context entities for reading or writing data by the workflow instances.
- **Activity trigger enhancement.** There is additional support for triggering an activity step to advance to the next step, even if the activity has not completed.
- **Use Symbols.** Define a variable for easy reference to an entity accessible from the workflow object using symbol. For example, `<Symbol name="AdverseParty" type="SubroAdverseParty" value="workflow.AdverseParty"/>`.

New and Changed in GScript in 5.0

IMPORTANT Guidewire renamed GScript to Gosu in ClaimCenter 6.0. Because this topic refers to ClaimCenter 5.0, it uses the term GScript, as that was the name in use for that release.

New in GScript in 5.0

The new ClaimCenter GScript features in this upgrade are:

GScript Annotations

GScript annotations are a simple way to provide meta-data about a GScript class, constructor, method or property. This annotation can control the behavior of the class, the documentation for the class. For more advanced usage, annotations can define an *interceptor* for a method. Interceptors allow you to wrap a method with code before and/or after the method runs. This enables certain types of *aspect programming*. For details, see “Annotations and Interceptors” on page 199 in the *Gosu Reference Guide*.

GScript Blocks

Create functions without names (*anonymous functions*) that you can define inline within another function. You can then pass that block of code to yet other functions to invoke as appropriate. Blocks are very useful for generalizing algorithms and simplifying interfaces to certain types of APIs, particularly things related to “find” type expressions or working with collections of objects. For details, refer to “Gosu Blocks” on page 213 in the *Gosu Reference Guide*.

GScript Generics

GScript generics let you describe collections of objects by the type of object they contain, similar to how this is done in Java 1.5. This feature allows abstractions of objects or functions so that they work with types of collections. However, at a compiler level, the function still has a certain type and is strongly typed at compile time. The result is that the function can work with many types of collections conveniently and type-safe at compile-time, and permit intelligent use of type inference to promote concise type-safe code. For details, refer to “Gosu Generics” on page 221 in the *Gosu Reference Guide*.

Publishing GScript Code as New SOAP APIs

The GScript language lets you write GScript code called from external systems as a web service using the SOAP protocol. Simply add a single line of code before the definition of a GScript class to publish it. For details, see “Web Services (SOAP)” on page 25 in the *Integration Guide*.

Calling SOAP APIs Published by External Applications

You can write code that easily imports SOAP APIs from external systems. By “subscribing” to the external web service and specifying it in Guidewire Studio, you can easily call external web services from GScript. For details, see “Web Services (SOAP)” on page 25 in the *Integration Guide*.

Native GScript Support for XML and XSDs

GScript can manipulate structured XML documents in two ways. You can use a tree of untyped nodes (*DOM-like untyped nodes*). Alternatively, read or create XML using an XML Schema Definition file (an *XSD file*) to produce a strongly-typed set of nodes. In both cases, GScript code can interact with the XML data as if nodes were native GScript objects. For details, refer to “Gosu and XML” on page 245 in the *Gosu Reference Guide*.

GScript Enhancements

GScript *enhancements* enable you to augment classes and other types with additional concrete methods and properties. Use this to define additional utility methods on a Java class or library that cannot be directly modified, or to extend GScript classes or Guidewire entities with additional behaviors. This is most useful when a class’s source code is unavailable, or a given class is *final* (defined explicitly not to be subclassed or otherwise overridden). Enhancements can be used with interfaces as well as classes. This enables you to add methods to interfaces in cases in which certain library designers were limited by their native language’s shortcomings. For details, see “Enhancements” on page 209 in the *Gosu Reference Guide*.

GScript List Enhancements: Sort, Find, Map, Each

Manipulating collections and lists uses Java’s collections and lists, however there are important differences because of GScript enhancements to collections. Enhancements are special GScript additions to a class that can add GScript methods to other classes. This is especially valuable to modify Java classes not originally defined with GScript features such as *GScript blocks*, which are anonymous in-line defined functions. These features allow concise and easy-to-understand GScript code that manipulates lists. Often with a single line of code, you can loop across items in the collection to perform complex tasks. For example: perform an action on each item in a collection, extract information from each item, or sort the collection. For details, see “Collections” on page 231 in the *Gosu Reference Guide*.

GUnit Testing

You can now unit test your GScript code, even complex code that requires a running ClaimCenter server. For more information, see “GUnit Tester” on page 182.

New GScript Try/Catch Syntax

GScript's try and catch syntax has new features for exception handling. In previous releases, multiple lines of code were required to catch and test for catch *checked exceptions*. Checked exceptions identify **specific** types of problems, typically thrown by Java code at some lower level. At a fundamental level, GScript does not natively distinguish between checked and unchecked exceptions. Guidewire recommends that you use standard unchecked exceptions for designing new code and APIs. However, the language now provides a simplified syntax for catching checked exceptions.

GScript provides a new syntax that lets you concisely catch only specific checked exceptions, similar to Java's try/catch syntax:

```
try {
    doSomethingThatMayThrowIOException()
}
catch( e : IOException ) {
    // Handle the IOException
}
```

GScript does not natively distinguish between checked and unchecked exceptions. However, when you definitely need to handle a specific exception, Guidewire suggests that you use this new simplified syntax to make your code more readable. The older syntax will continue to work in this release.

For more information, see “Try-Catch-Finally Constructions” on page 93 in the *Gosu Reference Guide*.

GScript Interface Support

GScript now has native support for interfaces, which are a set of required functions necessary for a specific task. For example, if a Guidewire application defines a plugin interface, in previous versions of GScript it would need to be implemented as a GScript class. In the current version, the interface can be implemented as a “real” interface natively in GScript. The advantage of this feature is that the GScript compiler can validate the required methods are present and that they have the appropriate method signatures. Interfaces are defined very similarly to GScript classes except that instead of writing the first line of the class definition:

```
class IFooInterface
```

...instead write:

```
interface IFooInterface
```

To use an interface, add “implements *MyInterfaceName*” after the class name:

```
class Bar implements IFooInterface
```

If a class implements more than one interface, separate the interface names by commas.

For more information about interfaces, see “Interfaces” on page 191 in the *Gosu Reference Guide*.

IMPORTANT The new interface features affect integration programmers, as discussed further in “GScript Plugin Important and Required Changes” on page 156.

GScript Inner Classes

GScript now supports for *inner classes*, both named inner classes and anonymous inner classes, similar to how they are used in the Java language. These are useful to encapsulate utility classes that are useful only the current class or to classes that derive from the current class. For detailed information, refer to “Inner Classes” on page 184 in the *Gosu Reference Guide*.

GScript Abstract, Override, and Final Modifiers

GScript now supports the modifiers `abstract`, `override`, and `final`, similar to how they are used in the Java language.

The `abstract` modifier indicates that a type is intended only to be a base type of other types. Typically an abstract type will not provide implementations (actual code to perform the function) for some or all of its functions and properties. This modifier applies to classes, interfaces, functions, and properties.

Apply the `override` modifier to a function or property implementation to declare that the subtype overrides the implementation of an inherited function or property with the same signature.

If GScript detects that you are overriding an inherited function or method with the same name but you omit the `override` keyword, you will get a compiler warning. Additionally, Guidewire Studio offer to automatically insert the modifier when it seems appropriate.

The `final` modifier applies to types, type members, local variables, and function parameters. It specifies that the value of a field, local variable, or parameter cannot be modified after the **initial** value is assigned. The `final` modifier cannot be combined with the `abstract` modifier on anything. This is because the `final` modifier implies that there is a concrete implementation and the `abstract` modifier implies that there is no concrete implementation.

For more information, refer to “Modifiers” on page 178 in the *Gosu Reference Guide*.

GScript Transaction APIs

GScript now provides APIs to let you manipulate what business data entities will be committed to the central database with the current action. For many tasks in GScript, you do not need to know how database transactions work because changes to the current object are automatically committed in the database as appropriate. Or, if errors occur that prevent the entire action from safely completing, the entire transaction will *roll back*, which means to undo any changes that were only tentatively made.

However, for some situations, you must understand database transactions. Sometimes you must add entities to the current database transaction, move entities from one logical transaction to another, or explicitly commit or roll back the transaction. For more information, see “Bundles and Transactions” on page 275 in the *Gosu Reference Guide*.

As a consequence of this change, the `EntityFactory` class is deprecated in GScript. Use the new bundle and transaction APIs to move a read only entity into the current bundle. For more information and code examples, see “EntityFactory Deprecated in GScript (Not in Java)” on page 139 and “Bundles and Transactions” on page 275 in the *Gosu Reference Guide*.

GScript Enumerations

GScript now natively supports enumerations, which define a list of codes, each with unique integer values associated with them. For more information, see “Enumerations” on page 189 in the *Gosu Reference Guide*.

Collections and String Size and Length Improvements

GScript adds enhancements for the `Collection` and `String` classes to support both the `length` and `size` properties, so you now can use the terms interchangeably with no errors. In previous releases, programmers were required to use different properties depending on whether the object was a collection or a `String` object.

List and Map Indexing Improvements

Both `List` and `Map` support indexed property access for both reading and writing in GScript.

For example, the following code gets `String` values from a list.

```
var strs = new ArrayList<String>(){ "a", "ab", "abc" }
strs.set(0, "b")
var firstStr = x.get(0)
```


You can write this instead in the more natural index syntax:

```
var strs = new ArrayList<String>(){ "a", "ab", "abc" }
strs[0] = "b"
var firstStr = x[0]
```

Similarly, the following code gets String values from a HashMap:

```
var strs = new HashMap<String, String>(){ "a" -> "b", "c" -> "d" }
strs.put("e", "f")
var valueForE = x.get("e")
```

You can write this instead in the more natural index syntax:

```
var strs = new HashMap<String, String>(){ "a" -> "b", "c" -> "d" }
strs["e"] = "f"
var valueForE = x["e"]
```

For more information, see “Collections” on page 231 in the *Gosu Reference Guide*.

List and Map In-line Initialization Improvements

There are new initialization styles for in-line defined elements of lists and maps.

For example, the following is an example List initializer that uses `java.util.ArrayList`:

```
var strs = new ArrayList<String>(){ "a", "ab", "abc" }
```

That is effectively shorthand for the following code:

```
var strs = new ArrayList<String>()
strs.add("a")
strs.add("ab")
strs.add("abc")
```

Also when defining list or map data and passing the data directly to a method, the type of the list can be inferred from the method signature. For example, instead of writing:

```
printStrings( new ArrayList<String>(){ "a", "ab", "abc" } )
```

...you can simply type:

```
printStrings( { "a", "ab", "abc" } )
```

For details of initializer improvements, see “Collections” on page 231 in the *Gosu Reference Guide*.

Improved Java Interopability

The GScript language can now more closely interact with Java. In previous releases, you could call Java classes.

In addition, in the current release you can also:

- Write GScript classes that extend from Java classes
- Write GScript interfaces (a new feature) that extend from Java interfaces, including with generics support
- Write enhancements in GScript that extend all instances of Java classes or interfaces, adding methods and properties to them. For more information, see “Enhancements” on page 209 in the *Gosu Reference Guide*.

Changed in GScript in 5.0

The major changes to Guidewire GScript in this ClaimCenter upgrade are:

GScript Constructor Syntax Changed

As in previous releases, GScript class can optionally have a *constructor*, which is a special function within the class that GScript calls when an instance of that class is created. For example, when GScript uses code like “`new MyClass()`”, GScript will call the class constructor for the `MyClass` class, which might initialize variables or perform other actions.

In previous releases, the syntax for a constructor was a method of the same name as the class. For example, if the class `MyClass` defined a `MyClass` method, GScript would call that method during object instantiation.

In the current release, the syntax for a constructor is to name the method simply `construct` rather than matching the class name. Additionally, omit the word `function`.

This the following is an example of the new constructor syntax:

```
class MyClass
{
    construct()
    {
        print("A MyClass object was just created!")
    }
}
```

The old constructor syntax is deprecated.

Guidewire Studio will attempt to detect when creating new classes with the old constructor syntax, and will offer to convert the class to the new constructor style. The old constructor syntax in your legacy code continues to work without problems in this release. However, the old style will be unsupported in a future release.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Convert GScript class constructors to the new constructor style.

Before your next upgrade, convert old-style GScript class constructors to the new style.

Stricter Return Statements Requirements for All Code Paths

In the current release, all possible code paths must return a value in a method that declares a return type. In other words, GScript requires a return statement for all possible paths through the method including all choices for conditional execution, such as `if` and `switch` statements. In previous releases, GScript allowed class methods that returned values to have code paths that did not return a value.

For example, the following method would be acceptable in previous releases but not in the current release:

```
class Bar {
    function MyFunction(myParameter) : boolean {
        if myParameter==1
            return true
        if myParameter==2
            return false
    }
}
```

GScript generates a “Missing Return Statement” error for this function and you must fix this error. Suppose you believe the function is always to be called with *myParameter* set to value 1 or 2 and that it returns a value in each case. The GScript compiler sees two separate `if` expressions for a total of four total code paths. To fix this, rewrite the method so that all code paths contain a return statement.

For example, the earlier example can be fixed using an `else` clause:

```
class Bar {
    function MyFunction(myParameter) : boolean {
        if myParameter==1
            return true
        else
            return false
    }
}
```

Similarly, if you use a `switch` statement, consider using an `else` section.

This stricter requirement for return statements mirrors the analogous requirements in the Java language.

✅ UPGRADE TASK

Priority: Before starting the server

Summary: Ensure you address all GScript syntax errors Studio indicates due to stricter requirements for all code paths.

Review your syntax errors that Studio flags for you and ensure you address any related to this change. Refer to the example above for how you might change code that causes issues.

Java Type Information Changes

Type information for Java classes now automatically includes properties for public fields and protected fields when there are no `get...` or `set...` methods to expose them.

There is no upgrade task associated with this change.

EntityFactory Deprecated in GScript (Not in Java)

Because of the new transaction APIs, the `EntityFactory` class is deprecated when used from GScript. Use the new bundle and transaction APIs instead to move a read only entity into the current bundle. It will continue to work in this release, but eventually you must change your code to the new APIs.

Specifically, instead of `EntityFactory` in GScript, use code such as the following:

```
uses gw.transaction.Transaction

...
// get current writable (mutable) bundle
var bundle = Transaction.getCurrent();

// add to current writable bundle
bundle.add(myEntity)
```

Note: As usual, create entirely new entities with the standard GScript `new` operator; the `EntityFactory.newEntity(...)` method is only appropriate when this class is used from Java.

For more information about the Transaction APIs, see “Bundles and Transactions” on page 275 in the *Integration Guide*.

✅ UPGRADE TASK

Priority: Before your next upgrade

Summary: Ensure you remove uses of the deprecated entity factory class in GScript. Java usages of the class do not need changes.

Before the next release, ensure you remove uses of the unnecessary (deprecated in this release) entity factory class in GScript. You do not need to change usages in Java because that version is necessary and not deprecated.

IMPORTANT The `EntityFactory` is deprecated from use in GScript but it is still appropriate for use from Java when used with the Java entity libraries. It is **not** deprecated when used from Java.

GScript Libraries Changes

Studio no longer supports the use of the library functions defined in the **Libraries** folder (either utility or class extension libraries). Instead, former library functionality exists now as either a GScript *class* or as a GScript *enhancement*. See “Library Functions Become Classes and Enhancements” on page 185 and “GScript Enhance-

ments” on page 134 for details.

UPGRADE TASK

Priority: Before starting the server.

Summary: No longer rely on library functions defined in the Libraries folder in Studio. These functions become GScript classes or GScript enhancements in this release.

GScript Case Sensitivity Implications

Starting in the current release, GScript will compile and run faster if you write all your GScript as case-sensitive code. Guidewire strongly recommends you always use proper capitalization precisely as defined for all names of types, variables, keywords (such as `var` and `if`), methods, properties, packages, and other language elements. Additionally, using proper capitalization makes your code easier to read.

For example, if an object has a `Name` property, do not write:

```
var n = myObject.name
```

Instead, use the code:

```
var n = myObject.Name
```

Similarly, use class names properly. Do not write:

```
var a = new address()
```

Instead, use the code:

```
var a = new Address()
```

Capitalization in the *middle* of a word (camel-case) is also important. Do not write:

```
var date1 = gw.api.util.DateUtil.currentdate()
```

Instead, use the code

```
var date2 = gw.api.util.DateUtil.currentDate()
```

UPGRADE TASK

Priority: Before deploying a production server

Summary: Review code for case issues, particularly for performance-intensive code. This results in faster GScript performance and lower memory usage.

Guidewire strongly recommends changing any existing code to be case-sensitive code, and writing all new code to follow these guidelines.

In a future release, case sensitivity will be strictly enforced, so you are encouraged to fix all case sensitivity issues now for performance reasons and future upgrade compatibility.

IMPORTANT Guidewire strongly recommends you write all GScript as case-sensitive for all type names, variable names, keywords, method names, property names, package names, and other language elements. If you do not, your code compiles slower, runs slower, and requires more memory at compile time and at run time.

Java Properties Capitalization Changes

From within GScript, Java properties are now capitalized to conform to GScript style standards of first letter being capitalized. There is no upgrade task associated with this change. If you use the new bulk fix tool in Studio to correct capitalization, it will use the new capitalization style.

Deprecation of itype Property

The `itype` property on objects and other types is deprecated. It represented the declared type (the type declared at compile time). However, this duplicated type-related methods that exist directly on the type, as well as the functionality of the more useful `typeof` operator.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Ensure you remove references to the `itype` property on objects and other types. Use the `typeof` operator instead.

Instead of using `itype`, use the `typeof` operator, which also existed in previous releases:

```
var f = "greetings"  
var t = typeof f
```

If you want information from a specific type known at compile time, simply access that type directly in GScript:

```
print(Integer.Name) // prints the String "java.lang.Integer"
```

In contrast, the `typeof` operator operates with the run time type of the object:

```
print((typeof 29).Name) // prints the String "java.lang.Integer"
```

For more information, see “Type System” on page 299.

Minor Change In Meaning of ‘List’

In previous releases, when using the type `List` in GScript code without fully qualifying it (typing it as `java.util.List`), GScript treated the type as `java.util.ArrayList`. Because the `List` class is an abstraction of a list and `ArrayList` is the common implementation of it, GScript performed the expected behavior in most real-world cases. This special GScript behavior occurred even if the script had a “uses `java.util.List`” statement above it.

In the current release, `List` resolves to `java.util.List` except when used in new expressions. In new expressions, `List` continues to be treated as functionally equivalent to `ArrayList`:

```
var strs = new List<String>(){ "a", "ab", "abc" }
```

This change will likely not affect most real-world code during an upgrade.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Ensure there are no syntax errors in Studio due to minor change in meaning of the Java list class in GScript code in unusual cases.

In the rare case that legacy code called the `clone` method on a list, it might generate a compile error due to this change. This is because the `clone` method exists on `ArrayList` but not `List`. This can be fixed by explicitly changing the code from `List` to explicitly mention `ArrayList`.

Typekey.Name Becomes DisplayName and UnlocalizedName

Typekeys, which are elements of typelists such as `AddressType`, had a `Name` property in previous releases. This property has been removed and replaced with two different properties: `DisplayName` for the name to display on the page (affected by locale settings) and `UnlocalizedName` (unaffected by locale settings).

IMPORTANT If you want a stable, unique, and unchanging `String` to represent the typecode, always use the `Code` property rather than any typekey name properties.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Convert all references to a typekey name property to the new localized name field. The localized name field displays in the local language. In rare cases, convert the typekey name property reference to the unlocalized name field, for debugging or compatibility with older code.

If you previously used the `Name` property, change to `typekey.DisplayName` for the localized language version of the display name or `typekey.UnlocalizedName` for the main name listed in the data model.

In almost all cases, use the `Code` field if you might store or compare values later on. Use the `DisplayName` if you are displaying something to the user, including notification emails or other messages in the local language. The `UnlocalizedName` property in this release contains what was the `Name` property in previous releases. However, this property mostly exists for compatibility reasons and is not generally useful since `Code` and `DisplayName` are the most useful.

To extract display name information, you can use code like:

```
var displayString = myCode.DisplayName
```

For example:

```
print(AddressType.TC_BUSINESS.DisplayName)
```

This code prints:

```
Business
```

If you have an extremely large number of uses of the `Name` property on typekeys, you could add `Name` property enhancement on the `Typekey` type to return the `UnlocalizedName` property. This would effectively let your legacy code continue to work and access the old property name. If you use this approach, use the appropriate property `DisplayName`, `UnlocalizedName`, or `Code` in any new GScript code that you write. For more information about enhancements, see “Enhancements” on page 209 in the *Gosu Reference Guide*.

If your application is multi-lingual and manipulates typekeys, choose very carefully whether you want to get the `DisplayName` property, the `UnlocalizedName` property, or the `Code` property.

IMPORTANT If you used the typekey `Name` property in previous releases, Guidewire strongly advises you to change your code to use the new name properties or the `Code` property as appropriate. In almost all cases, use the `Code` if you might store or compare values later on or use the `DisplayName` if you are displaying something to the user. Rarely, if ever, use the `UnlocalizedName`.

Type.TypeKeys Property Becomes Method with Argument

In previous releases, the type property `Typekeys` returned all typekeys, including *retired* typekeys as well as *unretired* typekeys. Since most Guidewire APIs use only *unretired* data and this property’s behavior was confusing in

some cases. The `Type.Typekeys` property works in this release but is now deprecated and will be removed in a future release.

✓ UPGRADE TASK

Priority: Before your next upgrade

Summary: Ensure all references to the deprecated `Typekeys` property of a type are converted to the new method syntax.

Remove references to the `type.Typekeys` property. Instead, call the new `getTypeKeys` method:

```
TYPENAME.getTypeKeys(includeRetiredTypekeys)
```

The `s` argument indicates whether to include retired typekeys in the returned list. If set to `true`, retired typekeys are included. Otherwise, will return only unretired typekeys.

For example, to return the same data as in previous releases, replace `MyType.Typekeys` with `MyType.getTypeKeys(true)`

TypeList.getByCode() Changes

The `getByCode` method of a `typelist` type behaves differently with respect to typecodes that are marked as *retired*. For example, consider this code:

```
CoverageType covType = CoverageType.getByCode(coverageCode);
```

In previous versions, this *ignored* retired typecodes.

In ClaimCenter 5.0, this *includes* retired typecodes.

If you want to filter out retired typecodes, you can check whether the new typecode is retired. From Gosu, check the `typekey.Retired` property, or from Java code use the method `isRetired()`.

For related information, see “Typekey.Name Becomes DisplayName and UnlocalizedName” on page 142.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: Review uses of the type list method `getByCode` to ensure you do not rely on ignoring retired typecodes.

Type Information Exposed for All Types

GScript now exposes all type information for all types within the code suggestion (dot completion) dialogs. In previous releases, not all types were supported, and not all information was exposed.

For example, if you type `String` followed by a period/dot character (.) in Studio code editor, you see much more information. Also, Java generics are fully supported for GScript classes and Java classes due to native support for generics. See “GScript Generics” on page 134.

There is no upgrade task associated with this change.

Removal of toMap Method on Collection

The collection enhancement method `toMap` is now unavailable. Guidewire reserves the right at a later time to implement a method with this same name but different behavior to match industry conventions for this method

name. Additionally, it is important that you do not create a replacement method with the name `toMap`. Guidewire reserves this method name on collections for future use.

WARNING Remove usages of `Collection.toMap(block)` from existing code. Plan for it to be unavailable in future releases. However, Guidewire reserves the right to reintroduce a method with the same name but different behavior in a future release.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Remove usages of the `toMap` method on collections. Do not reintroduce a method with the same name because Guidewire reserves that method name on collections for future use.

To upgrade your code, find existing code that uses the collection method `toMap`, for example:

```
var mapOfAToB = collectionOfA.toMap( \ a -> a.B )
```

Instead, build the result map directly without using the `toMap` method:

```
var mapOfAToB = new HashMap<A, B>()

for( a in collectionOfA ) {
    mapOfAToB[ a ] = a.B
}
```

If you have further questions about this change, contact Guidewire Customer Support.

Change in Equality Comparison, and New Warnings for Edge Cases

GScript now behaves more conventionally with respect to equality operations and implicit coercion. Equality operations are now typically symmetric with respect to the method used for testing equality. In other words, the method used to test if `A == B` is the same one used to test if `B == A`. As a consequence, there are fewer cases of implicit coercion between `A` and `B` in either direction. In some cases, such as when using implicit `String` coercion, Gosu now favors the left-hand-side operand's type to determine which internal method to use to compare the objects.

To reduce the chance that a change in behavior results, Gosu now detects and generates compiler warnings. Look for compiler warnings for potentially unintentional comparison between objects of different types in cases where the method used to test equality is asymmetric. The warning message has the following format:

```
"Asymmetric comparison between {0} and {1}. Consider casting one of the operands."
```

In the message, `{0}` and `{1}` contain the types of the left and right operands, respectively.

To avoid this situation, type the variables when you declare them, to ensure that the type is as you expect.

For example, the `getOriginalValue` method is declared to return the compile-time type `Object`, not typed specific to the exact typecode.

Suppose you have this code:

```
var priorCovStat = cov.getOriginalValue("MyCoverageStatusCde");
if (priorCovStat != "NR" && priorCovStat != null){
    count = count + 1;
}
```

This statement fails to evaluate properly if you do not declare the type of `priorCovStat` as the type `MyCoverageStatusCde`. That is because the typecode coercion from `String` to the typecode requires knowing which typelist to use, and at compile time the typecode is simply an `Object`.

The conventional declaration change is as follows:

```
var priorCovStat : MyCoverageStatusCde = cov.getOriginalValue("MyCoverageStatusCde");
```


However, the preferred Guidewire syntax is to use the “as” keyword for explicit conversion on the right side of the expression. Let GScript type inference define the type of the variable:

```
var priorCovStat = cov.getOriginalValue("MyCoverageStatusCde") as MyCoverageStatusCde
```

WARNING If you see compiler warnings for this issue, Guidewire strongly recommends that you fix the code to be more explicit to prevent the warning.

New and Changed in Installation in 5.0

New in Installation in 5.0

This topic describes what is new for installation of ClaimCenter 5.0. This includes:

- QuickStart Installation
- Support for WebLogic JNDI Data Source

QuickStart Installation

The ClaimCenter application comes bundled with a fully functional, in-memory database and application server for demonstration and development purposes. Using the directions in the QuickStart process, you are able to quickly view and develop your application with minimal steps. You can also use QuickStart to view and integrate ContactCenter with ClaimCenter.

Benefits include connecting to Guidewire Studio, editing configuration resources, and viewing the results immediately on the local server, without having to generate or deploy an EAR or WAR file. See “Installing a ClaimCenter Development Environment” on page 31 in the *Installation Guide* for details.

Support for WebLogic JNDI Data Source

You can configure ClaimCenter to use a JNDI data source managed by WebLogic. This provides a secure option for storing database credentials and allows you to manage the database connection pool through the application server. Previously, ClaimCenter could only use a JNDI data source managed by WebSphere. See “Using a JNDI Data Source” on page 48 in the *Installation Guide*.

Changed in Installation in 5.0

This topic describes what has changed for installation of ClaimCenter 5.0. This includes:

- SQL Server Parameter Required
- Updated JVM Heap Size Considerations
- Integration with ContactCenter Simplified

SQL Server Parameter Required

The SQL Server `READ_COMMITTED_SNAPSHOT` parameter must be set to on to deploy and start ClaimCenter. See “Configuring SQL Server for ClaimCenter” on page 23 in the *Installation Guide*.

Updated JVM Heap Size Considerations

JVM heap sizes have been updated to more accurately reflect both 32-bit and 64-bit applications. See “JVM Heap Size Considerations” on page 15 in the *Installation Guide* for details.

Integration with ContactCenter Simplified

Guidewire has simplified the integration process with ContactCenter and ClaimCenter using GScript. Previously, the integration code relied on external entities in ClaimCenter (`gw-entity-cc.jar`) and ContactCenter (`gw-soap-ab.jar`). These dependencies had to be regenerated whenever you modified or added fields to entities (such as through the `extensions.xml` file). Since GScript has full access to the Guidewire type system, the code can be viewed and if it has to be changed, no compilation is required.

ClaimCenter and ContactCenter communicate through SOAP APIs. ClaimCenter defines a web service (`abintegration`) which enables it to talk to ContactCenter through its published `IContactAPI`. ClaimCenter also defines a GScript plugin, `IAddressBookAdapter`, which basically invokes the integration layer. Using Studio, you configure the two applications by removing the demo adapter and uncommenting the real adapter.

You have the option of integrating ClaimCenter with ContactCenter using the QuickStart installation method or other servers and databases.

For details on integration, see “Integrating ClaimCenter with ContactCenter” on page 11 in the *Contact Management Guide*.

New and Changed in Integration in 5.0

New in Integration in 5.0

Major New SOAP Features in GScript

The GScript language lets you write web services (SOAP APIs) that external systems can call. Simply add a single line of code before the definition of a GScript class to publish it. You can also now write code that easily imports SOAP APIs published from external systems. By subscribing to the external web service and specifying it in Guidewire Studio, you can easily call external web services from GScript. For details, see “Web Services (SOAP)” on page 25 in the *Integration Guide*.

New Email APIs, Including Document Management

For more information, see the *ClaimCenter Rules Guide*.

EntityFactory in Java Method Name Change

When using the `EntityFactory` class in Java (which is not deprecated) or in GScript (a usage now deprecated, as described earlier), there is a new static method. The new static method is called `getInstance()`. It returns an instance of `EntityFactory`, which is necessary for calling some methods. This is intended to replace the `getEntityFactory()` method, which continues to work in this release but is deprecated. For example, the following code demonstrates this usage in Java:

```
EntityFactory.getInstance().newEntity();
```

Multicurrency Integration Support

To make financial transactions in multiple currencies, there are integration changes:

- ClaimCenter needs a way of getting current currency exchange rates. Integrate ClaimCenter with an external system to get this information by implementing the new exchange rate set plugin (`IExchangeRateSetPlugin`) interface. The main task of `IExchangeRateSetPlugin` is to create `ExchangeRate` entities encapsulated in an `ExchangeRateSet` entity. For more information, see “Exchange Rate Integration” on page 236 in the *Integration Guide*.
- Existing web services to import financials to a claim now have additional requirements on import related to multicurrency support. For details, see “Claim Financials Web Services” on page 200 in the *Integration Guide*.
- Similarly, for staging table import there are new currency-related requirements. See “Financial Transaction Importing Restrictions” on page 439 in the *Integration Guide*.
- New web service APIs for exchange rate adjustments, described in “Multicurrency Foreign Exchange Adjustment SOAP APIs” on page 202 in the *Integration Guide*.

See “Multiple Currencies” on page 171 in the *Application Guide* for more information about multicurrency.

SOAP API Non-Conversational Mode Now Supported

ClaimCenter now optionally supports non-conversational SOAP calls, which means that authentication happens for each API call, rather than authenticating once for each session. This enables fully-load-balanceable SOAP API calls, so that network administrators need not attempt to implement server session affinity (sometimes called session stickiness) in clustered systems. For more information, see “Conversational and Non-Conversational SOAP Modes” on page 62 in the *Integration Guide* and “Web Services Using ClaimCenter Clusters” on page 61 in the *Integration Guide*.

Rendering Arbitrary Input Stream Data, Such as PDF Data

You can display arbitrary `InputStream` content in a window. For example, you can display a PDF returned from code that returns PDF data as a byte stream (`byte[]`) from a plugin, encapsulated in a `DocumentContentsInfo` object.

Use the following utility method from GScript including PCF files:

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

Database Staging Table Import Conversion View

If you are implementing database import, you may want to use the **Conversion View** in the Data Dictionary. Get to the conversion view from the main **Home** page of the Data Dictionary, with the links that say:

- **Data Entities (Conversion View)**
- **Typelists (Conversion View)**

The **Conversion View** hides all fields that are not backed by actual database columns or are not importable in staging tables. For example, it hides all virtual fields which are calculated from other fields at run time when accessed from GScript.

IMPORTANT Use the **Conversion View** in the Data Dictionary to see the application data model from a conversion engineer perspective. It only shows fields backed by real columns that are also importable.

Messaging Threads Configurable

When you set up a new messaging destination in Studio, you can increase the number of threads the server uses for sending messages to that destination. You can also override this value later. For more information, see “Message Ordering and Multi-Threaded Sending” on page 170 in the *Integration Guide*.

Changes in Integration in 5.0

IDataObjectAPI is Deprecated

The web services interface `IDataObjectAPI` is now deprecated. Re-implement the core functions of this API by using the native GScript-SOAP interface to create custom APIs for your integrations. The old interface will continue to work for now but will go away in a future release.

Because you can now easily create custom APIs in GScript and expose them easily as web services, re-implement code using `IDataObjectAPI` to perform the specific tasks necessary. This will significantly improve performance for some types of integrations. It will also reduce complexity of your integrations by moving the **logic** of the integrations with external systems more into the product itself.

For example, in previous releases, a typical integration might have written a query expression passed to the `findIDs` method to get a list of public IDs. Then, the integration used `getObjectByID` to get each entity in the results, perhaps with an optional `ObjectFilter` object graph to reduce the number of fields retrieved over the network connection. Also, `getObjectByID` APIs require coercion from `GWObject` (the result of `getObjectByID`), which can make integration code harder to read and more error-prone.

In contrast, in the current release, you can rewrite the logic of the query **directly** in GScript within Guidewire Studio to use the following features:

- Use the power of Guidewire Studio syntax checking, compile-time verification, type safety, and code completion.

Design APIs that by design return only the data that you want to pass over the network.

UPGRADE TASK

Priority: By next release.

Summary: If you use the deprecated web service interface `IDataObjectAPI`, rewrite your integration code to define your own new web services customized for each integration point.

For APIs that take a public ID, write new APIs that return an entire entity across the network as a result from the web service. This will reduce the number of API calls, although in some cases is far too much data to transfer. For better performance, design new custom GScript classes within Studio that represent the true data model of your external system. Typically this means that it will have fewer fields than the full ClaimCenter entity. Then, write your own new web service to get and set instances of your custom GScript class.

For example, to return claim fields needed by a check printing system, create either a new GScript class or a custom data model extension entity called `CheckPrintIntegrationData`. Design this integration shell object to contain only the fields that you need. You can then create a new API that takes a public ID and returns a `CheckPrintIntegrationData` object.

IMPORTANT Using custom objects to transfer information for your custom web services is demonstrated in detail in “Publishing a Web Service” on page 32 in the *Integration Guide*. Also see “Testing Your Web Service” on page 39 in the *Integration Guide*.

The following table lists the methods of `IDataObjectAPI`, all of which are now deprecated:

Method	Description	Rewrite As...
<code>find</code>	Find a set of objects of the same type using the GScript finder notation	A GScript find statement within a new custom web service API defined in Studio.
<code>findIds</code>	Find a set of public IDs of the same type using the GScript finder notation.	A GScript find statement within a new custom web service API defined in Studio.
<code>getNextSequenceNumber</code>	Get the next sequence number for a sequence key	Logic that needs a new sequence number can be moved to logic within a new custom web service API defined in Studio. Within GScript you can use the code: <pre>n = gw.api.system.database.SequenceUtil.next(10, 1, "seqID")</pre>
<code>getObjectById</code>	Get an object graph by the public ID of the root object.	A GScript statement within a new custom web service API defined in Studio. If you previously passed null for the <code>ObjectFilter</code> argument, the new API would simply return an entity directly, such as <code>return a Claim</code> . If you passed an <code>ObjectFilter</code> to reduce the number of fields and subobjects, design your new API to return new GScript class instances containing all data the external system requires.
<code>getPathValues</code>	Return properties accessed from a root entity.	Within a new custom web service API defined in Studio, directly access the desired data using standard GScript path expressions.
<code>setPathValues</code>	Add, update, and remove values on entities relative to a root entity.	Within a new custom web service API defined in Studio, directly modify the desired data using standard GScript path expressions.

IMPORTANT For information and examples of creating custom web service APIs in GScript, see “Web Services (SOAP)” on page 25 in the *Integration Guide*. Particularly see “Publishing a Web Service” on page 32 in the *Integration Guide* and “Testing Your Web Service” on page 39 in the *Integration Guide*.

ObjectFilter and Related APIs Deprecated

As part of the deprecation of the SOAP interface `IDataObjectAPI`, the `ObjectFilter` class is also deprecated. It was a class mainly used with `IDataObjectAPI` to return only a sparse subset of an object graph.

UPGRADE TASK

Priority: By next release

Summary: If your integration code used the deprecated web service object filter class, avoid it by writing custom web services that download only data you need for each integration point.

In addition to `IDataObjectAPI` SOAP interface deprecation (see “`IDataObjectAPI` is Deprecated” on page 151 in the *Integration Guide*), the following APIs are deprecated because they rely on `ObjectFilter`:

- The ClaimCenter `IClaimAPI` interface’s method `getClaim`.
- ContactCenter’s `IContactAPI` interface’s method `retrieveContact`.

These APIs will continue to work in this release but may be unsupported in future versions.

You can write new GScript-based SOAP APIs that directly output **only** the fields you need, rather than letting the SOAP client determine which fields. If there are several different `ObjectFilter` types you used with previous releases for different integration contexts, write different SOAP APIs for each integration point. In other words,

define a new SOAP API method for each different subset of entity data. You can create custom GScript classes designed specifically for each integration point and your API can return one of these summary objects to the external system.

In the case of ContactCenter, when extracting contact information using custom SOAP APIs, write your API to only return the required fields and also only the required **relationships**. Relationships identify other contacts that are related, such as employees of a company, or a doctor-patient relationship. If you return an entire ABContact or unneeded related contacts, you will experience memory problems during SOAP serialization. See “Address Book Integration” on page 299 in the *Integration Guide* for more information.

For more information and examples of creating and exposing custom APIs in GScript, see “Web Services (SOAP)” on page 25 in the *Integration Guide*.

For more information and examples of creating and exposing custom web service APIs in GScript, see “Web Services (SOAP)” on page 25 in the *Integration Guide*. Particularly see “Publishing a Web Service” on page 32 in the *Integration Guide* and “Testing Your Web Service” on page 39 in the *Integration Guide*.

IMPORTANT If you used IDataObjectAPI or APIs that relied on ObjectFilter, those APIs are deprecated and will continue to work in this release. However, begin to re-write and re-factor your integration points to avoid these APIs. Instead, write custom SOAP APIs designed for each integration point rather than use these general APIs. As needed, create new GScript classes to encapsulate the required data for each integration point.

Plugins Directories Moved In Relation to Config Directory

The directories to put Java plugin files, Java plugin libraries, and Java files called from GScript have moved.

In addition to the new Studio modules architecture that affects all customer code locations, the `plugins` directory is no longer a **subdirectory** of the `config` directory within the module. The `plugins` directory is now at the same hierarchical level as (it is a peer of) of the `config` directory within each module.

Java Classes Called from GScript

The new locations for Java classes called from GScript are:

```
ClaimCenter/modules/configuration/plugins/GScript/classes
ClaimCenter/modules/configuration/plugins/shared/classes
```

The new locations for library JAR files (.jar files) used by those classes:

```
ClaimCenter/modules/configuration/plugins/GScript/lib
ClaimCenter/modules/configuration/plugins/shared/lib
```

Java Plugins

The new locations for Java plugin class files (.class files) is:

```
ClaimCenter/modules/configuration/plugins/pluginidir/classes
ClaimCenter/modules/configuration/plugins/shared/classes
```

The new locations for library JAR files (.jar files) used by those classes:

```
ClaimCenter/modules/configuration/plugins/pluginidir/lib
ClaimCenter/modules/configuration/plugins/shared/lib
```

The `pluginidir` placeholder in these paths refer to the `pluginidir` directory attribute in plugin registration.

UPGRADE TASK

Priority: Before starting the server

Summary: Confirm your Java classes were moved to the correct locations that changed due to the new modules feature. You must also update your Java development environment to generate Java class files or libraries to the new locations.

Confirm your Java classes were moved to the correct locations and then update your Java development environment to generate new class files or Java libraries to the new locations. If you use a Source Code Management (SCM) system, ensure your files are properly checked in and managed in the new locations.

Web Service Visibility and Customizability Changes

In addition to the new GScript features for publishing web services and using external web services, there are some other minor changes.

Web services are now documented in Studio by going to the **Help** menu and selecting **GScript API Reference** and looking for the interface's class name. This change does not require any upgrade task.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Optional. Review web service interfaces for security settings to determine whether to suppress any web services completely or change required permissions for a remote user to use to the service.

You could choose to completely hide some services for security reasons and make them unavailable from any external system. You can now disable any built-in web services by commenting out the `@WebService` annotation that publishes the service. You can also choose to

For more information, see “Publishing a Web Service” on page 32 in the *Integration Guide*.

UPGRADE TASK

Priority: Before next release

Summary: Optional. Review web service interfaces to determine whether to suppress generation in the toolkit for any web services for performance reasons of toolkit regeneration during development. This setting never affects run time performance nor run time security.

As in previous releases, all web service APIs by default appear in the generated Javadoc documentation in the toolkit after you regenerate the toolkit. This is the default both for included web services and for your custom web services. In many cases, this is the appropriate setting because most customers use generated Java libraries when connecting to web service APIs from external code. You can choose to publish a web service but suppress generation of the web service files in the toolkit if you do not need them. For example, if you use Microsoft.NET to connect to a ClaimCenter web service, it would typically use the dynamically-generated WSDL published by the ClaimCenter server rather than the toolkit-generated WSDL. In such cases, you would not need the Java libraries. Starting in this release, you can choose to override the toolkit generation behavior with an optional feature in the `@WebService` annotation to suppress toolkit generation. For more information, see “Publishing a Web Service” on page 32 in the *Integration Guide*.

WARNING Merely disabling generating the web service in the toolkit does **not** prevent outside code from calling this web service. If a web service is published, the server accepts connections if it has proper authentication credentials even if this parameter is set to `false`. To disable the web service, **comment out** the `@WebService` annotation entirely using the `//` symbols before that line. If you comment out this line, the class is not published the application will not accept new SOAP client connections to the service.

Regenerating the Toolkit Changes

The script in previous releases that regenerated the toolkit, `regen-toolkit.bat`, is no longer in the application's `bin` directory. In the current release, you must use the `gwcc` command line tool with the `regen-toolkit` parameter. There is no upgrade task other than to note the new procedure for regenerating the toolkit in this release.

To regenerate the toolkit:

Use the gwcc command line tool:

1. In Windows, bring up a command line window.
2. Change your working directory to:
`ClaimCenter/bin`
3. Type the command:
`gwcc regen-toolkit`

Plugin Registration Changes

In previous releases, plugins were registered in the central `config.xml` file. In the current release, they are registered using a special plugins editor within Studio.

UPGRADE TASK

Priority: Before starting server

Summary: Review the new Plugins editor in Studio and confirm your plugin settings upgraded properly. Additionally, if you use ContactCenter, there is a special required procedure.

The new Plugins editor user interface allows you define for each plugin interface what code implements it: GScript plugin, Java plugin, built-in plugin, or no plugin. For more information, see “Using the Plugins Editor” on page 141 in the *Configuration Guide*. In addition to the different user interface, the plugin registrations are now stored in a separate file from `config.xml`. Review the new Plugins editor in Studio and confirm your plugin settings upgraded properly.

If you used ContactCenter, you must additionally do the following:

1. In Studio, in the Plugins list, right-click on the `IAddressBookAdapter` and select **Delete**.
2. Confirm the deletion.
3. Right-click on `IAddressBookAdapter` and select **Implement** to create a new plugin implementation corresponding to ContactCenter.
4. Follow the directions in “Integrating ClaimCenter with ContactCenter” on page 11 in the *Contact Management Guide* to add the a GScript plugin to communicate with ContactCenter.

Plugin Built-in Implementation Classes Package Changes

Built-in implementations of plugins implemented GScript used to be in the hierarchy:

`Plugins.DIRECTORY.CLASSNAME`

They moved to to:

`gw.plugin.DIRECTORY.impl.CLASSNAME`

This applies only to the classes of built-in implementations. The package and class names of the interfaces have not changed.

UPGRADE TASK

Priority: Before starting the server

Summary: If you used built-in demonstration plugins or default plugin implementations, review your code for package names that may have changed.

Depending on which plugins you register that use built-in demonstration plugins or default implementations, this might affect your initial installation and configuration during upgrade.

This change typically would not affect your integration source code. The **interface** package namespace for the plugins did not change, only built-in implementations of some plugins and you might not be using them anyway. If you do not update your plugin configuration correctly for this change, you will see an error message in the console log when starting up the server.

Messaging Destination Registration Moved

In addition to the new plugin editor, the messaging registry is no longer in the main `config.xml` file. Instead it is now in a separate file managed through a new messaging editor in the Guidewire Studio interface. See “Using the Messaging Editor” on page 161 in the *Configuration Guide* for details of the messaging registration editor. The new destination registry contains the same information as in the previous release, but is moved to a separate file and edited through an editor rather than as raw XML. There is no upgrade task for this change.

GWServices Plugins Removed

In previous releases, ClaimCenter-ContactCenter communication was handled through a special type of plugin called a GWServices plugin. ClaimCenter no longer has a special type of plugin just for inter-application communication. In the current release, if you use both ClaimCenter and ContactCenter, configure the `IAddressBookAdapter` plugin interface as a normal GScript plugin with class `ccabintegration.CCAddressBookPlugin`.

For standard authentication to ContactCenter, add plugin parameters `username` and `password` in the plugin registry user interface. Alternatively, you might want to use an authentication plugin for unusual authentication or to hide authentication information in a separate file. If so, add any desired plugin parameters for your authentication plugin, and then add the parameter `authPlugin` with value of `true`. The built-in `IAddressBookAdapter` implementation that connects to ContactCenter will use the specified authentication plugin. For more information about authentication plugins, see “ABAAuthenticationPlugin for ContactCenter Authentication” on page 246 in the *Integration Guide*.

For a related upgrade task for ContactCenter users, see “Plugin Registration Changes” on page 155.

GScript Plugin Important and Required Changes

GScript now has a native implementation of *interfaces*, which is a defined set of required methods necessary to perform some certain task. For more information about interfaces, see “GScript Interface Support” on page 135. Guidewire requires customers convert any GScript plugins (plugins defined in GScript) to use GScript interfaces to get proper compile-time validation. The language itself confirms that you implemented the required interfaces correctly. Additionally, if in future releases if a plugin interface changes, Studio can easily identify that the interface does not match the customer implementation of the interface. For example, if a new method is required to be implemented, Studio will flag this error if your code does not match.

All plugin interfaces are now accessible to the GScript world in a different package hierarchy than in previous releases. The location of the plugin class is much more important in the current release since the plugin implementation class must reference the plugin interface explicitly. This is further discussed in the next section about interfaces. All plugins are now defined in the package `gw.plugin.*`, so as soon as you type “`gw.plugin.`” in your GScript code, Studio will display your plugin choices.

UPGRADE TASK

Priority: Before starting the server

Summary: Change older GScript plugins to implement a GScript interface.

When implementing a Guidewire-provided plugin interface, you must change the line that defines the GScript class. In previous releases, GScript classes that define plugins would simply look like a regular GScript class, as follows.

```
class FooTransport {
  ...
}
```

In this release, you must specifically declare what interface the plugin claims to implement, such as:

```
class FooTransport implements gw.plugin.MessageTransport {
  ...
}
```

Making this change may reveal compilation errors if your class does not properly implement the plugin interface. You must fix these issues.

A common compilation issue is that a method that implements interface methods that look like properties must be implemented in GScript explicitly as a GScript property. In other words, if the interface contains a method starting with the substring "get" or "is" and takes no parameters, you must define the method using GScript property syntax. In contrast, do not define it as a simple method. For example, if plugin `APPlugin` declared a method `getBar()`, your plugin implementation of this interface looks like:

```
class MyAPPlugin implements gw.plugin.APPlugin {
  property getBar() : String {
    ...
  }
}
```

UPGRADE TASK

Priority: Before starting the server

Summary: Change older GScript plugins to use new-style plugin parameter syntax.

Some customer-written GScript plugin implementation used parameters passed from `config.xml`. If any of your plugins used this feature, you must make minor changes to use these parameters because the application passes parameters in a different way to GScript plugins.

Note: Plugins are no longer registered in `config.xml`. In this release, plugins are registered directly in Studio, which stores them in a different file (see “Plugin Registration Changes” on page 155). To change values of these parameters, use Studio to edit them.

In previous releases, parameters were passed to GScript plugins using a Map passed to the class constructor:

```
class MyAuthenticationSourceCreator
{
  MyAuthenticationSourceCreator( params : java.util.Map )
  {
    ...
  }
}
```

In the current release, there are two tasks you must do to use parameters.

First, implement the initializable plugin (`gw.plugin.InitializablePlugin`) interface in addition to the plugin interface you want to implement. Note that classes can implement more than one interface; simply separate the implemented interfaces with commas. Because your class implements the initializable plugin interface, the application knows to call the `setParameters` method on your class.

Next, simply implement the `setParameters` method (instead of the class constructor), which takes a Map of your parameters passed in from the plugin registry you defined in Studio. The Map is the same name/value format as in previous releases.

The following example demonstrates how to define an actual plugin that uses parameters:

```
uses java.util.Map;
uses java.plugin;

class FooTransport implements MessageTransport, InitializablePlugin {
```

```
// note the empty constructor. The application now longer calls the constructor
// with a map of parameters. If you do provide an empty constructor, the application
// will call it when the plugin is instantiated, which is before setParameters is called
construct()
{
}

function setParameters(map: java.util.Map) {

    // here we can access values in the MAP to get parameters defined in plugin registry in Studio
}

// NEXT, define all your other methods required by the MAIN interface you are implementing...

function suspend() {}

function shutdown() {}

function setDestinationID(id:int) {}

function resume() {}

function send(message:entity.Message, transformedPayload:String) {
    print("=====")
    print(message)
    message.reportAck()
}
}
```

UPGRADE TASK

Priority: Before the next release

Summary: Optional. Review Java plugin implementations to determine if some might be better as GScript plugins to take advantage of new GScript and Studio features, such as native web service support.

Because of new native GScript support for calling external web services and a native implementation of interfaces, some customer plugins written in Java can be easily rewritten in GScript. Simply rewrite it as a GScript plugins. For example, if your Java plugin simply made a SOAP call to a remote system, you can do this natively in GScript in the current release.

For more information on SOAP, see “Web Services (SOAP)” on page 25 in the *Integration Guide*. For more information about the interface changes, see “GScript Interface Support” on page 135.

There might be a very small number of tasks impossible from GScript but possible in Java, although this number is likely smaller than you think. For example, a GScript plugin can call out from GScript to a pre-existing Java library or classes that implement communication protocols to a legacy system. You might be able to write much of your plugin in GScript in these cases. If desired, you can still call out to Java code from GScript as needed for a small number of tasks or to call legacy code. For more information on calling Java, see “Overview of Calling Java from Gosu” on page 102 in the *Gosu Reference Guide*.

Changing some Java plugins to GScript may improve your ability to maintain your plugin code. For example, it might reduce the frequency you need to manually recompile Java code and regenerate the Java toolkit.

SOAP Plugins No Longer Needed

Although in previous releases, GScript and Java plugins were the recommended ways to implement plugins, customers could optionally implement plugin interfaces as *SOAP plugins*. This was necessary because there was no way to directly call a remote SOAP API from GScript. If a customer registered a SOAP plugin instead of a GScript plugin or Java plugin, ClaimCenter the web service across the network. Due to new GScript features in this release, SOAP plugins are not needed as a separate type of plugin and is no longer a native option for plugin implementations.

UPGRADE TASK

Priority: Before starting the server

Summary: If you used SOAP plugins (plugin implementations that call a remote SOAP service), convert them to GScript plugins that call out to your SOAP API for each method.

If you implemented a SOAP plugin in a previous release, convert it to a GScript plugin that calls out to a remote SOAP API as needed. Your new plugin could optionally contain different method signatures than the ClaimCenter-defined plugin interface.

To convert a SOAP plugin

1. In Studio, register the legacy system's SOAP web service as a new SOAP endpoint.
2. Write a new GScript plugin (a plugin implemented in GScript) that implements the desired interface.
3. For each method in the GScript plugin, use GScript's native SOAP features to call the remote legacy server's relevant SOAP API and return results as needed.

IAssignmentAdapter Deprecated

The assignment plugin (IAssignmentAdapter) is deprecated. This plugin let you encapsulate assignment logic in a plugin rather than defined in rule sets in Studio. Guidewire recommends that you convert any previous assignment plugin to code either in rules or in GScript classes that encapsulate your assignment logic and that can be called by rules.

UPGRADE TASK

Priority: Before the next release

Summary: If you use the deprecated assignment plugin, rewrite your code to instead use the new assignment APIs.

There are assignment methods on assignable objects that allow you to write intuitive code that assigns objects. For example, assignable objects now have methods like `assignActivityToQueue`, `assignByUserAttributes`, `assignGroup`, `assignGroupByGroupType`, and `assignGroupByLocation`, `assignGroupByRoundRobin`, `assign` (which takes a User and Group), and `assignManually` (takes a User), and many others. In addition, you can call `entity.autoAssign` to assign the item using the rules to determine the group and user assignment.

For example, the following code assigns a user using a standard round robin algorithm:

```
myClaim.assignUserByRoundRobin(true /* include subGroups */)
```

As in the previous release, these methods can be called on assignable objects from assignment rules. In addition, in this release you can call these methods from any GScript, not just assignment rules. For example, you can call these methods from Event Message rule sets or PCF files.

The deprecated plugin will continue to work in this release. However, the set of assignments that are possible will not include assignment types available in domain methods directly on the assignable entities.

IExportTools Removed

The IExportToolsAPI web service interface was removed. This interface contained a method to export an entire dataset from administrative database tables in an XML format to transfer it to another database or to re-import it. The amount of data transferred over the web services protocol could be too large because of the size of datasets.

UPGRADE TASK

Priority: Before the next release

Summary: If you use the deprecated export tools web service interface, do one of the following instead. Either use the user interface to export your administrative data, or use a custom web service API that sends necessary data one record at a time.

The easiest way to export this data is to use the built-in user interface in the application. When logged into ClaimCenter with an administrative user, first click the **Administration** tab, then from the left sidebar, click **Import/Export Data**, then in the right pane, click **Export**.

If for some reason you must export administrative data programatically, you can design a custom web service to export your desired administrative information one record at a time. Do not export all administrative data in a dataset at once.

Geocoding Plugin Changes

ClaimCenter and ContactCenter include a supported geocoding plugin to connect to the Microsoft MapPoint geocoding service. The source code to this plugin is provided in the product, as in previous releases, for you to examine or customize. In previous releases, the included plugin was written in Java and included as an example in the developer toolkit. In the current release, it is written in GScript, and you can view the GScript plugin's code within Guidewire Studio.

The geocoding plugin interface added new features and also now supports a more strongly-typed approach for data passed between the application and the plugin. The new version of this interface is called `GeocodePlugin`, which is slightly different from the interface name in previous releases, `IGeocodePlugin`. The old interface is still present in the current release, as a deprecated interface. See the following section “If You Customized Geocoding Plugins” on page 161 for compatibility information for plugins written for previous releases.

The new `GeocodePlugin` interface must be implemented in GScript. It cannot be implemented in Java.

UPGRADE TASK

Priority: Before starting the server

Summary: If you used the included MapPoint plugin unmodified, reconfigure your server to use the new included MapPoint plugin that is implemented in GScript and has a different class symbol.

If you used the included MapPoint plugin in previous releases and you did not modify it, in this release simply use the new included MapPoint plugin. See “Geographic Data Integration” on page 315 in the *Integration Guide*.

Using the New Built-in MapPoint Plugin

The new version of the MapPoint plugin is defined in GScript. To configure it, see “Using the Geocoding Feature” on page 22 in the *System Administration Guide*. Note that to change the timeout value for connection with MapPoint, change the **Timeout** setting in the MapPoint web service definition in the Studio Web Service editor.

Geocoding Plugin Interface Improvements

The new geocoding plugin interface supports the following new features:

- Getting an address from latitude-longitude coordinates, a feature sometimes called *reverse geocoding*.
- Getting maps for arbitrary addresses.
- In driving directions, optionally return an overview map.

See “Geographic Data Integration” on page 315 in the *Integration Guide* for details.

Geocoding Plugin Differences in Address Correction

In addition to the plugin interface changes, there are differences in how ClaimCenter and ContactCenter handle address corrections during geocoding. These changes are independent of which plugin interface you use. In previous releases, the application assumes blank fields in a returned address are unchanged or uncorrected fields. In the current release, blank fields in a returned address are assumed to be blank purposely.

For example, address fields in return data might be unknown or inappropriate for that geocode status. If the geocode status is `City`, a blank street address might be appropriate because the returned coordinates do not represent a specific street address. In this context, the blank field is not exactly a change but is a clarification of the field value for that granularity (street address, city, postal code). The full set of fields returned by the geocoding plugin are assumed to be the complete set of fields to show to the user. They are also assumed to be the set of fields that define what to log to the geocoding corrections table.

UPGRADE TASK

Priority: Before deploying a production server

Summary: If you used the included MapPoint plugin and relied on old behavior of blank address fields in geocoding results in any PCF pages, update your code for the new behavior.

If any of your PCF pages or other code relies on the old-style geocoding behavior for blank fields in the address for geocoded addresses, update your code accordingly.

If You Customized Geocoding Plugins

As mentioned earlier, the geocoding plugin interface added new features and changed to support a more strongly-typed approach for data passed between the application and the plugin. To support implementations that use the older interface, Guidewire created the new interface with a slightly different name (`GeocodePlugin`) and left the old plugin interface (`IGeocodePlugin`).

UPGRADE TASK

Priority: Before starting the server

Summary: If you customized geocoding plugin Java files, either merge your changes as GScript code into the new geocoding plugin or use an included temporary (until next upgrade) plugin wrapper.

If you customized the older Java plugin example for MapPoint or wrote custom Java or GScript implementation of the geocoding plugin interface, rewrite your code to use the new interface.

Alternatively, you can use a **temporary wrapper mechanism** to support older plugin code. It will continue to work in this release, but the temporary mechanism is deprecated and will go away in a future release. Guidewire encourages you to begin rewriting any custom geocoding plugin interface to directly use the new geocoding plugin interface.

In all cases, ClaimCenter and ContactCenter only calls plugins registered to the **new** interface, which is called `GeocodePlugin`. This release includes a special built-in wrapper plugin implementation that allows you to use your old plugin that uses the old interface, which is called `IGeocodePlugin`.

Register your plugin implementation for the old interface and register the built-in wrapper plugin for the new interface. The wrapper plugin modifies method arguments and then calls your implementation of the old interface. The basic geocoding and driving directions functionality still works. However, you will not get any new geocoding-related features such as supporting finding addresses from latitude-longitude coordinates, or getting maps for arbitrary locations.

Independent of the plugin interface you use, there are differences in how ClaimCenter and ContactCenter handle corrections. See “Geocoding Plugin Differences in Address Correction” on page 161 for details. If any of your PCF pages or other code relies on the old-style behavior, you must update your code accordingly.

To use a geocode plugin written for a previous release

1. In Studio under **Resources**, right-click **Plugins**.
2. Select the menu item **New** → **Plugin**.
3. In the dialog that appears, choose the `IGeocodePlugin` interface. This is the old interface.
4. In the right pane, click **Add...** and select Java or GScript from the picker as appropriate for your old plugin implementation.
5. In Studio under **Resources**, right-click on **Plugins** → **GeocodePlugin** and choose **Delete**. This deletes the configuration for the included GScript-based MapPoint plugin that implements the new interface.
6. Studio under **Resources**, right-click on **Plugins**.
7. In the dialog that appears, choose the `GeocodePlugin` interface. This is the new interface.
8. In the right-pane, click **Add...** and select Java from the picker.

Note: Generally speaking, the new `GeocodePlugin` interface must be implemented in GScript. It cannot be implemented in Java. This Java support for this interface is only supported when using the built-in wrapper plugin to call an old interface implementation.

9. In the **Class** field, enter the class name of the built-in wrapper plugin implementation:
`gw.plugin.geocode.IGeocodePluginWrapperPlugin`
10. You do not need to modify any other fields. For example, you can leave the plugin directory field blank.

IMPORTANT Alternatively, if you modified the older included MapPoint plugin, make code modifications in the latest version of the included plugin implementation. The new MapPoint plugin uses the new plugin interface and is now written in GScript.

ISO Configuration Changes

This release changes how to configure ClaimCenter to use the built-in support for Insurance Services Organization (ISO).

In previous releases, you would enable the built-in ISO support by setting the `config.xml` parameter `EnableISOMessageSink` to `true`. In the current release, the `EnableISOMessageSink` parameter is no longer supported.

In previous releases, you could add or remove plugins or messaging transports in the `config.xml` file. In the current release, this information is managed in other editors in Studio, and enabling the plugins and messaging transport also the new way to enable ISO support.

There is no upgrade task for this item. If `EnableISOMessageSink` was set in your `config.xml` before upgrading, this setting is automatically set for you.

To enable the built-in ISO support

1. In Guidewire Studio in **Resources**, click **Messaging**.
2. Click the row with ID 66 and the name `iso`.
3. Check the **Enabled** checkbox.
4. Within Studio, click the **Plugins** node and expand it.
5. Click the `isoTransport` node.
6. Check the **Enabled** checkbox.

Document Management Changes

Document Template Descriptors Optionally Cached

By default, the list of document templates is calculated from files locally on disk when the application needs them. For small lists of document templates, this is a quick process. However, if you have a large number of document templates, you can tell ClaimCenter to cache the list for better performance.

You might prefer to use the default behavior (no caching) during development, particularly if you are frequently changing templates while the application is running. However, for production, set the new optional parameter in the document template source plugin to cache the list of templates.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Optional. If you have many document templates, consider enabling the new document template descriptor caching feature for better application server and user interface performance.

To enable document template descriptor caching

1. In Guidewire Studio, under **Resources**, click **Plugins** → **IDocumentTemplateSource**.
2. Under the parameters editor in the right pane, add the `cacheDescriptors` parameter with the value `true`.

Note: This parameter name is case sensitive.

Document UUIDs Caching When Validation Errors Occur

In previous releases, there could be unexpected behaviors if a new document was created and a validation error occurred when the Validation rule set runs within the same database transaction. The validation error ensures the database transaction rolls back, which means that no database data was changed. However, if the document was created in the external system and the local ClaimCenter transaction rolls back, there is no stored reference in the ClaimCenter database. The database does not contain the document unique ID (UUID) that describes the document's location in the external system. This information is stored in the Document entity in ClaimCenter in the same transaction so it is not committed to the database. The new document in the external system is effectively orphaned, and additional attempts to change ClaimCenter data will re-generate a new version of the document.

For the common case of validation errors that are soon fixed, application behavior changed in this release. When a validatable entity fails validation, ClaimCenter saves the document UUID in local memory. If the user fixes the validation error in that user session, the document information is saved and stored as expected so no externally-stored documents will be orphaned.

There is no upgrade task associated with this change.

WARNING It is still the case that errors that cause the transaction to roll back (such as uncaught GScript exceptions), an externally-stored document associated with the current transaction might be orphaned. The document would be stored but no ClaimCenter Document entity committed to the database will reference the document UUID for it.

Templates and Descriptor File Locations Changed

Template files and descriptor files changed locations due to Studio changes for modules and the new checksum actions for each module. The new location is:

`ClaimCenter/configuration/config/resources/...`

If you do not need some of these descriptor files, you can delete files as appropriate from that location in the configuration module. This applies to document templates, note templates, and email templates.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: To disable additional document templates or descriptors added to your configuration module, delete associated files in that module.

Document Management Plugin Files Directory Change

The built-in document content source plugin and metadata source plugin have associated files, as in previous releases. In this release, the location they store their files changed due to module-related changes in Studio.

In this release, by default (if a special parameter is absent) a temporary directory is used with file relative paths, and this actual location will vary by web application container. For Tomcat, files will be in:

`Tomcat_Root\work\Catalina\localhost\cc`

The DocumentMetadataSource built-in plugin does not actually allow the relative path to be configured, and document metadata is always stored within a metadata directory under this directory.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: Verify the document management directory settings used by content source and metadata source plugins.

If a web application directory relative path is used, ClaimCenter displays a warning on the console. Existing files will **not** be moved automatically. You must ensure that any required files are moved.

For production servers, Guidewire recommends that you do not use the built-in plugins. These plugins are intended as examples only, as discussed further in “Document Management Overview” on page 249 in the *Integration Guide*. However, for tests in production systems using the built-in plugins, specify a fixed full path for the directory rather than use a web application container relative path. To set this, set the parameter `documents.path` in the plugin configuration in Studio for both plugins to a fixed full file path. If you specify an absolute path, then for existing files no migration is necessary.

Message State and Message History Changes

Once a message has been acknowledged or skipped, either directly or through resynchronizing a claim, it is moved to a message history table. You will no longer be able to see this message in the administration console. The *pending retry* message state no longer exists.

There is an upgrade database trigger that marks pending retry messages as pending send messages, and another trigger that moves all messages that are inactive to the message history table.

✓ UPGRADE TASK

Priority: Before deploying a production server

Summary: Ensure your reporting tools do not rely on the removed message state called pending retry.

If you previously used any reporting tools to analyze messaging based on these inactive messages, you must change your code to point to the message history table. If you had any reports that looked for *pending retry* messages, instead look for *pending send* messages.

Messaging Transaction Changes and MessageRequest Plugin

If you use the optional `MessageRequest` plugin, be aware of slightly different rules about database transaction boundaries when sending a message. In previous releases, there were not clear transaction boundaries between the calls to `MessageRequest.beforeSend(...)`, `MessageTransport.send(...)` and then finally `MessageRequest.afterSend()`.

In the current release:

1. ClaimCenter calls `MessageRequest.beforeSend(...)` in one database transaction and commits changes assuming no exceptions occurred.
2. ClaimCenter calls `MessageTransport.send(...)` and then `MessageRequest.beforeSend(...)` in one database transaction and commits changes assuming no exceptions occurred.
3. As in previous releases, the `MessageReply` plugin, which optionally handles asynchronous acknowledgments to messages, does its work in a separate database transaction and commits its changes assuming no exceptions occurred.

At the start of each transaction, ClaimCenter locks the message and the primary object of the messages (if a primary object exists). This ensures that the application does not try to acknowledge a message that is already skipped, or similar conditions.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Confirm your messaging plugins do not rely on old transaction behavior between different phases of messaging plugin method calls, particularly in error and exception handling.

If you relied on specific transaction behavior (bundle commits) between different phases of messaging plugin method calls, confirm that your code conforms to the new behavior. Pay particular attention to how you handle errors and exceptions in all your messaging plugins.

Ensure Messaging Code Does Not Rely on Assignment Entity

The `Assignment` entity went away in ClaimCenter 5.0 as part of the larger improvements to the assignment system in ClaimCenter.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Ensure your integration code does not rely on the assignment entity, which is now gone. Instead use the new assignment data model and APIs to detect changes to assignments directly on assignable entities.

As part of upgrade, ensure you do not have dependencies on this entity, including but not limited to messaging code that relies on `Add`, `Changed`, `Removed` events on that entity.

If you have messaging code that relies on these events, you must rewrite `Event Fired` rules accordingly.

Starting in ClaimCenter 5.0.2, you can detect assignment events directly on assignable entities. For example, your `Event Fired` rules can be listed for `AssignmentAdded`, `AssignmentChanged`, and `AssignmentRemoved` events on assignable entities such as claims, exposures, activities, and matters. Note that the root object for the message will be the assignable entity, not the `Assignment` entity, which no longer exists.

Message Sinks Now Unsupported

In previous releases, message sinks were deprecated in favor of new messaging plugins such as `MessageTransport`, `MessageRequest`, and `MessageReply`. Customers were urged to rewrite message sinks with the new messaging architecture and new messaging plugin interfaces. In this release, message sinks are fully unsupported.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Ensure you do not use old-style message sinks, converting to new messaging plugins if necessary.

Ensure you do not use any old-style message sinks, converting to to new messaging plugins such as `MessageRequest`, `MessageTransport`, and `MessageReply` as appropriate.

Transaction Public ID Length Must Be Shorter

In addition to multicurrency-related integrity checks during staging table import, the transaction's `PublicID` field must be two or more characters shorter than the maximum field length.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Confirm your transaction public ID field is has appropriate length, at least two or more characters shorter than the maximum length.

Multi-Registration of Plugins for Wrapping Unsupported

In previous releases, you could wrap a plugin written in GScript around another registered plugin written in Java, where both plugins were registered to the same interface. The application would only call one implementation for that interface (the GScript version).

Other than the exceptions listed below, do not register multiple plugins for the same plugin interface. Support for this feature was deprecated, and there is no support for this feature in the Studio plugin configuration. However, if you used this feature in earlier releases, automatic upgrade handles it and it will work in this release, but avoid this approach and refactor your code.

If you want your GScript plugin to call out to Java code, it is still supported. You can continue to use the features discussed in “Overview of Calling Java from Gosu” on page 102 in the *Gosu Reference Guide* to call Java classes or libraries. You can even send and receive Guidewire entities using the tools discussed in “Using Java Entity Libraries” on page 117 in the *Gosu Reference Guide*. However, do not register your additional Java code in these cases *as plugins* in the plugin registry in Studio. Simply refer to your Java classes directly from your GScript code.

Exceptions To Multi-Registration Rule

The three interfaces for messaging plugins support multiple implementations of those interface. For example, you can register multiple `MessageTransport` plugins to talk to multiple external systems using the events and messaging system. To distinguish them, when you create the plugin in Studio, Studio will prompt you for a name for the plugin. You will use that plugin *name* when configuring the messaging destination in the Messaging editor in Studio.

✓ UPGRADE TASK

Priority: Before starting the server

Summary: Ensure you do not use multi-registration of plugins, which was rarely used and is now unsupported.

Contact Autosync

Address books can use web service to for automatic contact autosync functionality using a web service `IContactAutoSyncAPI`. When a contact is updated in an address book, it can notify ClaimCenter to pick up the change. ClaimCenter finds the contacts to synchronize to the new values stored in the external address book. If you use ContactCenter and have it installed, ContactCenter will use this automatically.

For more information, see “Address Book Integration” on page 299 in the *Integration Guide*.

Changes to Parameter Order in Outgoing SOAP API Requests

In ClaimCenter 5.0.4, Guidewire changed how the GScript language accesses remote web services registered in Studio. Due to this fix, for some remote SOAP API methods, the parameter order of APIs might change from the perspective of the GScript SOAP client code.

ClaimCenter now fixes parameter order so it always uses the parameter order in the method signature in the WSDL. To avoid unexpected behaviors upgrading from previous releases, Guidewire provides a new configuration setting. To use the behavior of previous releases, in the `config.xml`, set the configuration parameter `SortWSDLsUponStudioImport` to the value `true`. This value enforces backward compatibility for previous releases of ClaimCenter.

UPGRADE TASK

Priority: Before starting the server

Summary: Set the `config.xml` file parameter `SortWSDLsUponStudioImport` to `true` to enforces backward compatibility with how GScript handles remote web services registered in Studio. Set it to `false` for the new behavior, but that requires careful review of APIs that call remote web services for possible change in function argument order.

If you set this parameter to `false`, the order of arguments in some cases change for method signatures calls to remote SOAP API methods. You must review all outgoing SOAP API calls to confirm the correct parameter order before the next major release. However, you can defer this work until the next major upgrade by using the new configuration parameter `SortWSDLsUponStudioImport` with the value `true`.

WARNING Not all affected methods can be detected from compile errors. For example, if a method takes two instances of the same type as function arguments. If the argument order changes, then there is no compile error.

Setting `SortWSDLsUponStudioImport` to `true` works around a compile error in the included MapPoint implementation. If you set this parameter to `false`, then see the following section about MapPoint, “MapPoint Updates for New WSDL Parameter Ordering” on page 168.

If you are *not* upgrading from a previous release of ClaimCenter, then Guidewire recommends that you disable this setting (set it to `false`). This ensures that no conflicts occur with later major upgrades.

If you set `SortWSDLsUponStudioImport` to `false`, then review calls to remote SOAP APIs from GScript to confirm whether parameter order changed for any methods.

Studio updates types and API method signatures when it loads or refreshes the WSDL. To refresh the WSDL, click **Refresh** or **Edit** in the Studio Web Services editor.

To change the WSDL parameter order setting:

1. Set the `SortWSDLsUponStudioImport` parameter in the `config.xml` file. If you are upgrading from a previous release, then set to `true` for maximum backward compatibility.
Set it to `false` for a new project, or if you are upgrading from a release earlier than ClaimCenter 5.0.
2. Open Studio. Access each web service and click **Refresh**.
3. Verify the correctness of that web service:
 - a. In Studio, run the **Verify All Resources** command.
 - b. Search for usages of the API stub class corresponding to that particular web service.
 - c. Inspect each operation web service method call to ensure that method parameters are in the correct order. As mentioned before, not all affected methods can be detected from compile errors.

MapPoint Updates for New WSDL Parameter Ordering

In ClaimCenter 5.0.4, Guidewire changed how the GScript language accesses remote web services registered in Studio. The included MapPoint implementation that calls to the remote web service now works with either value of the new configuration parameter `SortWSDLsUponStudioImport`.

Note: For more information about this important change, see “Changes to Parameter Order in Outgoing SOAP API Requests” on page 167

If you set `SortWSDLsUponStudioImport` to `true`, which is recommended for upgrades, do not make changes to the included MapPoint implementation.

If you want to use the correct (fixed) parameter order, set the value of `SortWSDLsUponStudioImport` to `false`. You get compile errors within MapPoint code, but you can update MapPoint to work with this setting. To update the MapPoint implementation to work with the fixed parameter order (that is, with `SortWSDLsUponStudioImport` set to `false`). The change mentioned below fixes the compile errors.

UPGRADE TASK

Priority: Before starting the server

Summary: If you want to use the new WSDL behavior with correct (fixed) parameter ordering, make one small change to the built-in MapPoint implementation to avoid a compile error. This change is unnecessary if you set the value of `SortWSDLsUponStudioImport` to `true`, which enforces backward compatibility.

To support the fixed method parameter order:

1. Set the `SortWSDLsUponStudioImport` parameter in the `config.xml` file to `false`.
2. Open the Studio Web Services editor and select the **MapPoint** web service.
3. Click **Refresh**.
4. Navigate to the GScript class `gw.plugin.geocode.impl.MappointGeocodePlugin`.
5. Modify line 175 to the following:

```
var route = routeService.CalculateSimpleRoute ( startAndFinish, dataSourceName, segPref );
```

Changes to Message Ordering and Multi-Threading

ClaimCenter changed how it pulls messages from the send queue and dispatches them to the messaging plugins to send. In older releases and the current release, ClaimCenter pulls messages from the database (the send queue) in batches of messages on the batch server only. Next, it waits for a polling interval before querying again.

However, the details of how ClaimCenter queries for messages and dispatches them to messaging plugins changed in this release.

You can configure the number of messages that the messages subsystem retrieves in each round of sending. This is called the chunk size, and you configure this in the messaging destination configuration editor in the **Chunk Size** field. By default, this value is set to 100,000, which is typically easily includes all sendable messages currently in the send queue. You can also change the polling interval in messaging destination configuration editor in the **Polling Interval** field.

For the following documentation, note some important terminology:

- Messages that are claim-specific are called *safe-ordered messages*. All such messages have a claim or a claim subobject as the root object (their primary object).
- Messages that are not claim-specific are called *non-safe-ordered messages*. For example, messages whose root object is a Catastrophe entity are non-safe-ordered messages.

To understand the changes in this release, you must understand the difference between message readers and message sending threads:

- *Message readers* are threads that query the database for messages. Message readers use the *message send order* (typically this is equivalent to creation order). The message reader never loads more than the maximum number of messages in the chunk size setting at one time.
- *Message sender threads* are threads that actually call the messaging plugins to send the messages. A new feature in this release is support for multiple sender threads per messaging destination for safe-ordered messages. You can configure the number of sender threads for safe-ordered messages in the messaging destination configuration editor in the **Number Sender Threads** field.

In version 4.0, ClaimCenter reads messages as follows:

1. In 4.0 there is just one message reader thread. The message reader thread queries the database for the next batch of messages by creation order. It gets a list of messages irrespective of which messaging destination each message is for. The maximum number of messages retrieved at once is the chunk size.
To get the next batch of messages, ClaimCenter queries the database searching for a range of *send order values*, which are number value in each message's `SendOrder` property. You can think of the send order property as equivalent to message creation order. ClaimCenter automatically creates this value. In version 4.0 ClaimCenter uses the send order in the query itself.
2. The message reader thread sorts the list of messages by the send order.
3. The message reader thread distributes messages to each destination. Each messaging destination has its own single sender thread. Similarly, each messaging destination's sender thread has an associated queue of messages just for that destination. Each destination thread stays active and continues to send messages in its own destination-specific queue. Each destination constantly pulls the next item from its own queue and dispatches them one at a time to the messaging plugins to send.
The destination sender thread ensures that no two messages are *in-flight* (sent but unacknowledged) for the same destination for the same claim. It does this by checking before sending a message whether that claim has any unacknowledged messages for that destination. Internally, the destination actually segregates the queue of incoming messages by claim and processes them one at a time. ClaimCenter does continue to the next message in the queue for each claim until the previous one for that claim is acknowledged.
4. After the message reader thread distributes all of the original batch of messages to destination-specific queues, the message reading thread goes to sleep for the **remainder of the polling interval**. Suppose the amount of time since the last beginning of the polling interval is *TIME_PASSED* milliseconds, and the polling interval is *POLLING_INTERVAL* milliseconds. If there is still time before the polling interval completes, the reader sleeps for (*POLLING_INTERVAL* - *TIME_PASSED*) milliseconds. If the time passed is greater than the polling interval, the thread does **not** sleep before requerying. Note that the destination-specific sending threads continue to send while the message reader thread sleeps.

5. *For the message reader thread:* when the message reader awakens, it requeries the database for new messages. As the message reader finds new messages, it distributes all new messages by destination to each destination's queue. Remember that these are separate threads, so the destination-specific queue might be empty or increasingly large as time goes on.

IMPORTANT The fact that this in-memory queue can in some cases continue to grow without limit is one of the big differences between the architecture of version 4.0 and 5.0.

For destination sender threads: after a destination sends all possible messages in its queue, it waits for the message reader to add messages to its queue or a message acknowledgement. A message acknowledgment allows the destination to send other pending messages for a claim. The in-memory queue includes a possibly-large number of messages that are held up due to other messages waiting for an acknowledgement for a message for that claim.

6. If any message acknowledgements occur, ClaimCenter notifies the destination for that message so that the next held-up message for that destination for that claim can now send.

In version 5.0, the messaging ordering and sending architecture works as follows:

1. Each messaging destination has a worker thread that queries the database for messages for that destination only. In other words, *each destination has its own message reader*. Each message reader thread (each worker thread) acts independently.
2. The destination's message reader queries the database for one batch of messages. In contrast to earlier versions, in version 5.0 ClaimCenter does not use the send order in the query itself. Instead, ClaimCenter orders the results by send order. Each destination thread uses the chunk size to determine when to stop iterating across the query results after it retrieves that number of messages from the database.

ClaimCenter performs **two separate queries**:

- a. First, ClaimCenter queries for non-claim-specific messages, also known as **non-safe-ordered messages**. If the chunk size is not set high enough, the returned set is not the full set of non-claim-specific messages.
- b. Next, ClaimCenter queries for claim-specific messages, also known as **safe-ordered messages**. The maximum number of messages returned for this query is also the chunk size. (The chunk size is not cumulative for safe-ordered and non-safe-ordered messages.)

The database query itself ensures ClaimCenter that no more than one message for a specific claim is part of this list. If a message for that claim is sent but unacknowledged, it does not appear in this list yet. This enforces the rule that no more than one message can be in-flight for each claim per destination.

The query ensures that no two messages are in flight (sent but not yet acknowledged) for the same destination for the same claim. So, if there are 100 messages for one claim, the query only reads and dispatches **one** of those messages (out of a possible 100) to the destination subthreads.

WARNING By default, the chunk size is set to 100,000, which is usually sufficient for customers. Be sure not to lower the chunk size too much. Typically there are dependencies between safe-ordered and non-safe-ordered messages on that destination. If the chunk size is too low, the first query might not retrieve all the non-safe-ordered messages that your safe-ordered messages rely upon. For example, a claim message might reference a catastrophe. Thus, the downstream system probably needs ClaimCenter to send the catastrophe message before the claim message that references the catastrophe.

3. For each destination, the worker thread iterates through all non-safe-ordered messages for that destination, sending one at a time in a single thread to the messaging plugins.
- After each worker thread finishes sending non-safe-ordered messages, it creates subthreads to send safe-ordered messages for that destination. Configure the number of threads in the messaging destination con-

figuration editor in the **Sender Threads** field. Each worker thread distributes the list of safe-ordered messages to send to the subthreads.

IMPORTANT Assigning the number of sender subthreads for a destination affects only the safe-ordered messages for that destination. All non-safe-ordered messages always send in a single thread for each worker, which ensures that the `SendOrder` property dictates the send order for non-safe-ordered messages.

If the message is associated with a claim, during messaging operations you can optionally lock the primary entity (the `Claim`) at the database level. This can reduce some problems in edge cases in which other threads (including worker threads) try to modify objects associated with this same claim. ClaimCenter checks the `config.xml` parameter `LockPrimaryEntityDuringMessageHandling`. If it is set to `true`, ClaimCenter locks the primary entity during message send, during all parts of message reply handling, and while marking a message as skipped.

4. The message reader thread waits until all destination threads send all messages in the queues for each subthread.
5. ClaimCenter checks how much time passed since the beginning of this round of sending (since the beginning of step 2) and sleeps the remainder of the polling interval. Configure the polling interval in the messaging destination configuration editor in the **Polling Interval** field. Suppose the amount of time since the last beginning of the polling interval is `TIME_PASSED` milliseconds and the polling interval is `POLLING_INTERVAL` milliseconds. If there is still before the polling interval completes, the reader sleeps for $(POLLING_INTERVAL - TIME_PASSED)$ milliseconds. If the time passed is greater than the polling interval, the thread does **not** sleep at all.

The message reader reads the next batch of messages. Begin this procedure again at step 2.

IMPORTANT In version 5.0, the polling interval setting critically affects messaging performance. If the value is low, the message reader thread sleeps little time or even suppresses sleeping between rounds of querying the database for more messages.

To illustrate the integration changes, compare the following situations.

First, suppose there are two messaging destinations and the send queue contains 10 messages for each destination. For each destination, assume that there is no more than one message for each claim. In other words, for each destination, there are 10 total messages related to 10 different claims:

- In version 4.0, assuming the number of messages does not exceed the chunk size, the message reader reads all 20 claim-specific messages in the first database query. All messages (10 per destination) send in one thread per destination, one at a time. The message reader sleeps for the polling interval after querying the database and dispatching the messages to each destination. In version 4.0, note that the length of the destination queue could grow to unlimited size because the message reader is independent of the destination sender threads. The number of pending messages (waiting for previous messages for that claim) is unlimited.
- In version 5.0, assuming the number of messages does not exceed the chunk size, each destination gets only one message for the claim for that destination from the database. In this case, every message can be sent immediately because each message for that destination is independent because they are for different claims. If the destination's **Number Sender Threads** setting is greater than 1, ClaimCenter distributes all claim-specific messages to multiple subthreads. In this case, with more sender threads you might see higher performance than in version 4.0. The length of the destination queue never exceeds the number of messages queried in each round of sending. The message reader waits until all sending is complete before repeating.

In contrast, compare the old and new behavior if the messages for one destination includes 10 claim-specific messages for the same claim and 5 non-claim-specific messages:

- In version 4.0, assuming the number of messages does not exceed the chunk size, ClaimCenter reads 10 claim-specific messages and the non-safe-ordered messages in one database query.

- In version 5.0, assuming the number of messages does not exceed the chunk size, ClaimCenter reads all 5 non-safe-ordered messages and sends them. However, ClaimCenter only gets **one** message for the claim for that destination from the database. If the destination's **Number Sender Threads** setting is greater than 1, ClaimCenter distributes all claim-specific messages in multiple threads per destination. In this case there is only message, so ClaimCenter handles fewer messages per claim for each polling interval. In this case, you might see lower performance than in version 4.0 if you did not yet change other settings such as the polling interval or chunk size.

To improve performance, particularly cases like the second example, change the following settings in the Messaging editor for your destinations:

- Lower the **Polling Interval**. The value is in milliseconds. Experiment with lower values perhaps as low as 1000 (which means 1 second) or even lower. Test any changes to see the real-world effects on your messaging performance. If your performance issues are primarily related to **many messages per claim** per destination, then this is the most important setting to change.
- Increase the value for **Number Sender Threads**. This permits more worker threads to operate in parallel on the batch server. Again, test any changes to see the real-world effects on your messaging performance. If your performance issues are primarily related to many messages but **few messages per claim** for each destination, then this is the most important setting to change.

New and Changed in Rules in 5.0

New in Rules in 5.0

Guidewire has added the following new rules in ClaimCenter 5.0:

- Archive Rules

Archive Rules

Archiving is the process of moving claims from the primary ClaimCenter database into a secondary database for storage. You may still search for, retrieve and work with archived claims, but while archived, they are a much smaller burden to the primary database.

Guidewire has added new Archive rules that work in conjunction with two new batch processes to implement this functionality. The two new batch processes are `markforarchive`, which flags eligible claims for archiving and `archive`, which performs the archiving on flagged claims.

Note: For additional information on claim archiving, see “Archiving” on page 87 in the *Application Guide*. For additional information on how ClaimCenter uses the Archive rules, see “Archive” on page 48 in the *Rules Guide*.

Changes in Rules in 5.0

The major changes to business rules in ClaimCenter 5.0 are the following:

- Assignment Rules

Assignment Rules

Guidewire has made a number of significant improvements to assignment rules and how ClaimCenter invokes these rules:

- Guidewire has improved the way that round-robin assignment works. In ClaimCenter 5.0, you can use a set of criteria to construct the set of potential assignees, which can span groups. The criteria, not group membership, are important. However, if the criterion is the user's load factor, then the assignment methods still restrict candidates to those from a single group.
- There are new assignment methods. This include the `assignUserbyAttributeandLocation` and the `assignGroupbyLocation` methods.
- You may make immediate assignments through the use of the new `autoassign()` method.
- It is possible to write your own GScript in PCF pages in ClaimCenter 5.0 to call assignment rules and methods directly, bypassing the assignment engine.
- Guidewire provides new "dynamic" assignment functionality with new methods to create your assignments. Using this new functionality, you can better reflect your own business logic. For example, in ClaimCenter 5.0, it is easier to select users across groups, or to create and make assignments based on your own measures of work load.
- It is possible to assign extension entities in ClaimCenter 5.0.

Changes to the *assignUserByProximityWithSearchCriteria* method

There are subtle differences between ClaimCenter 4.0 and ClaimCenter releases prior to 4.0 in how the application handles the assignment of a user through a proximity search:

- In releases prior to 4.0, ClaimCenter looked first at the proximity criteria, regardless of group affiliation. It then filtered out users that were not members of the correct group. Thus, the method in it ally returned users that were not part of the assigned group.
- Starting with the 4.0 release, ClaimCenter first determines the users in the assigned group, then filters by proximity criteria. Therefore, the method returns only users that are in the assigned group, and then narrows the selection.

For example, suppose that you have the following group hierarchy:

```
Root
+-- GroupA
+-- UserA (in LA)
+-- UserB (in Irvine)
+-- GroupB
+-- UserC (in LA)
+-- UserD (in Irvine)
```

You then perform an assign-by-proximity search with the following criteria:

- A user with an address in LA
- A large proximity circle (or no restriction on the distance)
- The number of results capped at two
- An assigned group of GroupB

In releases prior to ClaimCenter 4.0, the `assignUserByProximityWithSearchCriteria` method returned both UserA and UserC. It then filtered out UserA as that user was not in *GroupB*. As a consequence, the method assigned UserC in every single instance. Starting in ClaimCenter 4.0, the method returns UserC and UserD (both are in GroupB). It then assigns each in turn using round-robin assignment. It is important to understand that if you call this method repeatedly, the method can return a different user each time. (That is, if there is a sufficiently large enough pool of valid users.)

Assignment Events

If the `AssignmentStatus` changes on an assignable object, ClaimCenter fires an assignment event. There are three possible events:

- `AssignmentAdded`
- `AssignmentChanged`
- `AssignmentRemoved`

There are several major changes in how ClaimCenter 5.0 handles assignment events in rules that check for assignment events:

- An assignment `CloseDate` change in ClaimCenter 5.0 (for example, skipping an Activity) triggers an assignment event.
- In assignment rules, the root object is the entity itself, rather than the assignment entity.

Guidewire has not changed the event names, simply their behavior.

Exposure Get Recoveries API Returns Read-only Entities

In ClaimCenter 4.0, the exposure method `getRecoveriesIterator(false)` returns recoveries that are already in a writable bundle. In ClaimCenter 5.0, you must manually copy this object to a writable bundle.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Review your code for use of exposure get recoveries API, and ensure you add the recoveries to a writable bundle.

New and Changed in Studio in 5.0

New in Studio in 5.0

The major changes to Guidewire Studio in ClaimCenter 5.0 involve the following areas:

- Standalone (Disconnected) Operation
- Studio Resource Management
- Studio Interface Changes
- New Configuration Options
- Graphical Studio Editors
- GUnit Tester
- New GScript API Documentation Support
- Keyboard Shortcuts
- New Contextual Right-Click Menu Commands
- Delete (or Revert to Base) Files
- View the SCM Log
- View and Edit (More) Files in Studio
- New Text Editor
- New Help Menu PCF Format Reference

Standalone (Disconnected) Operation

With the release of ClaimCenter 5.0, Studio has two connection modes of operation:

- connected, in which Studio accesses (reads and writes) files on the application server file system
- disconnected, in which Studio accesses files in the local file system

Studio starts in disconnected mode by default. (Studio indicates its connection mode in the lower right-hand corner of the screen.) To connect to the server, select **Connect to Server...** from the Studio File menu.

Studio Resource Management

With release of Guidewire ClaimCenter 5.0, Guidewire introduces a number of significant changes and modifications to the ClaimCenter file structure. Studio manages and displays all resource files (for example, PCF files, data model extension files, display key files, images, and GScript classes) and ClaimCenter application files.

ClaimCenter stores these files in module folders or directories, which it evaluates in a specific order at application startup. These directories may contain distinct copies of the same resource file. In that case, the predominate copy would be in the first one found, with Studio disregarding any others it finds after the first.

Studio organizes and displays these files as a “virtual” directory in the **Resources** pane. Even though the files exist in multiple directories, you see a single, flat, directory in Studio.

After you open a file from within Studio, Studio does one of the following:

- If this is the first time that you have opened the file, Studio asks you if you want to open a copy for editing.
- If this is *not* the first time that you have opened the file, Studio checks out your copy of the file for editing (if using a SCM).

You only work with your editable copies of the base configuration files. In this way, Studio stores all modified files in the same location (in the `/modules/configuraton` application directory). In the **Resources** tree, Studio indicates edited files by coloring the file name blue. It also colors red the names of files *not* checked into an SCM (if you have a SCM system associated with Guidewire Studio).

You can also access (view) a number of read-only base configuration files in the Studio **Data Model Extensions** → **metadata** folder.

Studio Interface Changes

With the release of ClaimCenter 5.0, Guidewire introduces a number of new icons in the Studio interface. For example, Studio uses icons to indicate the different file type and uses a different icon for editable files, non-editable files, and currently-being-edited files.

New Configuration Options

With the release of ClaimCenter 5.0, Guidewire has added a number of new configuration options that you access from **Tools** → **Options**. These include:

- Source Control Management (SCM) Integration
- Font and Color Selection for GScript Code
- “Smart Fix” Code Completion Options
- Connecting to Remote Server
- Linking Studio Files to an External Editor

Source Control Management (SCM) Integration

As of the release of ClaimCenter 5.0, Studio provides support for the following source control management systems:

- Perforce (existing)
- CVS (new)
- Generic (through the File Filter Adapter Module, which you can use to “skip” control files)
- None (through the File System module, which is the default if you do not actively set a SCM system.)

You configure how Studio interacts with the source control management system through the **Version Control** dialog (**Tools** → **Options** → **Configuration Settings**).

Note: If using a source control management system, Studio manages all file checkouts for you. To check a file into the SCM system, you must still manually check it into the SCM application.

Font and Color Selection for GScript Code

With the release of ClaimCenter 5.0, Studio provides a means to set the font type and size of GScript code. (Select **Tools** → **Options** → **Configuration Settings** → **Colors & Fonts**.) You can also set how Studio displays specific GScript code items, such as keywords or operators. Studio displays a code sample at the bottom of the dialog that reflects your settings so that you can view the effect of your choices immediately.

“Smart Fix” Code Completion Options

Along with the existing ability to enable/disable the display of deprecated methods in GScript code, Guidewire has added a number of automatic “smart fixes” for various code completion issues. You access this functionality from the **Tools** → **Options** → **Configuration Settings** → **Code Completion** dialog. You can enable/disable each of the following through this dialog:

- Add import for unrecognized symbol
- Make implicit coercion explicit
- Fix Java-style type cast
- Create unrecognized display key
- Convert string literal to new display key
- Change old constructor syntax to new
- Remove unused variable
- Add missing “override” modifier
- Implement functions and properties

For more information, see “Setting Code Completion Options” on page 97 in the *Configuration Guide*.

Connecting to Remote Server

With the release of ClaimCenter 5.0, the default Studio development environment is that of *stand-alone* mode. This means that by default, Studio only reads and writes to a local file system and only interacts with a local (non-production) application server. You can easily connect to a non-local server, however, by entering the server URL, user name, and password in the connection dialog. (You access the connection dialog by clicking the **Disconnected** icon in the bottom right-hand corner of the screen or by selecting **Connect to Server** from the **File** menu.)

You can use the **Tools** → **Options** → **Configuration Settings** → **Remote Server** dialog to enter the default server URL shown in the connection dialog. (Enter the fully qualified domain if the application server and the local file system are not physically located on the same machine.) You can also enter a value in the **Connect Script** field that defines the file location of a script to use during the connection process. Use this script to set connection parameters, for example.

Linking Studio Files to an External Editor

With the release of ClaimCenter 5.0, Studio manages the ClaimCenter resource files for you, and displays most of these files in a Studio editor dedicated to that purpose. However, there are some resource files that you can see in the Studio **Resources** tree, but for which there are no dedicated Studio editors. You can, if you want, link an external editor to a file type through Studio. Thus, when you attempt to open one of these files in Studio, Studio automatically opens the file in the linked editor. You set this functionality in the **Tools** → **Options** → **Configuration Settings** → **External Editors** dialog.

In this same dialog, you can link Studio to an external file *difference* tool (such as Araxis Merge). If you link to such a tool, you can use it to display the differences between two different versions of a file. You access this functionality by selecting **Diff Against Base** from the right-click menu (after selecting a file).

Graphical Studio Editors

The ClaimCenter 5.0 release of Studio includes a number of graphical editors that you can use to manage a number of Guidewire ClaimCenter resources. Many of these resources you previously defined and managed through XML files. The new editors include:

- Display Key Editor
- Entity Names Editor
- LOB Editor
- PCF Editor
- QuickJump Editor
- Typelist Editor
- Web Services Editor
- Workflows Editor

Display Key Editor

With the release of ClaimCenter 5.0, Studio incorporates a **Display Key** editor. (A `DisplayKey` represents a single user-viewable—and therefore, localizable—text string.) Using the editor, you can directly manage your display keys. This includes:

- Create, view, and edit display keys, for all locales
- Find all the usages of a display key in the application

You can also “hot” create a display key from within a GScript editor by entering a string literal, then pressing ALT+ENTER.

Entity Names Editor

With the release of ClaimCenter 5.0, Studio incorporates an **Entity Names** editor. (An entity name is a text string suitable for viewing in the ClaimCenter interface.) Using the editor, you can define GScript expressions for an entity display name. The editor consists of two parts:

- A table in which you manage symbols for use in the display name GScript code
- A text box for entering the GScript code that defines the display name

LOB Editor

With the release of ClaimCenter 5.0, Studio incorporates a graphical interface to edit the six special typelists that define the Lines of Business in ClaimCenter. These typelists are:

- | | |
|--------------|-------------------|
| • LossType | • CoverageType |
| • LOBCode | • CoverageSubtype |
| • PolicyType | • ExposureType |

As you make edits in the LOB editor, Studio maintains referential integrity within the parent-child relationships in the LOB structure. This means that if you make changes to a typecode's code or delete a typecode, then Studio update the typecodes that refer to that one. For more information, see “Using the Lines of Business Editor” on

page 583 in the *Configuration Guide*.

IMPORTANT However, if you modify one of these typelists through the Studio editor, you may still need to modify other typelists or PCF files that reference your modified typelist. This is because the **Lines of Business** editor *does not* edit other typelists, GScript code, and PCF files that refer to these six typelists. In some cases, this can lead to error states.

PCF Editor

With the release of ClaimCenter 5.0, Studio incorporates a graphical PCF (page configuration file) editor. Using the Studio PCF editor, you can modify an existing PCF file or add a new PCF file and graphically build and manage its elements. Studio displays a list of all PCF files as a tree structure in its **Resources** pane. Selecting a file opens a graphical representation of it in the middle pane. A separate right-hand pane contains the **Toolbox**, which contains the elements that you can insert into the page. The PCF editor also contains a property sheet for PCF elements that you use to edit their attributes. You can edit most element attributes directly. However, as of the release of ClaimCenter 5.0, you can no longer modify the PCF ID attribute.

QuickJump Editor

The **QuickJump** box is a text-entry box for entering navigation commands using keyboard shortcuts. Guidewire locates the box at the upper-right corner of each ClaimCenter screen. You set commands through the **QuickJump** configuration editor. To remove the **QuickJump** box from the application screens, remove all configuration commands from the **QuickJump** configuration editor.

Plugins Editor

With the release of ClaimCenter 5.0, Studio supports managing plugins directly through the Studio interface, within the **Plugins** editor.

Typelist Editor

With the release of ClaimCenter 5.0, Studio incorporates a typelist editor. (Guidewire calls the list of available values for a drop-down field a typelist.) Using this editor, you can do the following:

- Create new typelists
- Edit existing typelists
- Add, remove, and edit typecodes
- Define static type filters and dynamic key filters
- Manage localization of typecodes

Web Services Editor

With the release of ClaimCenter 5.0, Studio incorporates a **Web Services** editor to create and manage web services within Studio. Using this editor, you can:

- Expose a web service to GScript
- Display documentation for the service.

You use this editor to set the following web service parameters:

- URL
- Name
- Timeout (in seconds)
- Authentication mode

After completing the definition, Studio populates an area at the bottom of the editor with the JavaDoc associated with this web service.

Workflows Editor

With the release of ClaimCenter 5.0, Studio incorporates a **Workflows** editor. Using the editor, you can:

- Create and edit workflow definitions
- Arrange workflow in a graphical layout

Studio represents workflow as a hierarchical tree structure (in the middle pane) and as a set of graphical elements (in the right-most pane). As graphical elements, Studio represents each XML workflow element as a rectangular box with lines interconnecting the various elements. This indicates the flow of control from one element to the next. It also labels the various XML elements with icons (in both the hierarchy tree and the graphical editor) to indicate their purpose.

GUnit Tester

With the release of ClaimCenter 5.0, you can run tests of GScript code from within Studio by creating specialized GUnit test classes. Studio uses the QuickStart server to run the GUnit tests. Running a test opens additional panes at the bottom of the screen that show the tests to run and the results of each test. If a test fails, Studio provides error messages to help in debugging the problem.

Studio also provides a dialog box in which you can set the server runtime parameters to use for a test.

New GScript API Documentation Support

With the release of ClaimCenter 5.0, Guidewire provides API reference material for all GScript classes in the ClaimCenter base configuration that exist in the Studio **Classes** folder. This does *not*, however, include GScript classes that exist in the **Tests** folder.

Keyboard Shortcuts

Guidewire has added a number of keyboard shortcuts (keystrokes) that provide important code completion, navigation, and editing capabilities with the release of ClaimCenter 5.0. For a complete list, consult the *Guidewire Studio Guide*.

New Contextual Right-Click Menu Commands

Guidewire has modified the Studio right-click menu (accessed by selecting a resource, then performing a mouse right-click) and added additional new functionality. These new commands primarily add the following functionality:

Diff Against Base	You must link Studio to an external "difference" tool (such as Araxis Merge) to use this command. After selecting a file, use this command to view the differences between different versions of the file.
Source Control	You must link Studio to an external source control management (SCM) system to use this command. Selecting this command opens a number of submenu commands involving options related to managing Studio files in a SCM system.
Find in Explorer	Use this command to navigate to the selected file in the Windows Explorer window.

Delete (or Revert to Base) Files

On the Studio **Edit** menu, you see a **Delete** or a **Revert to Base** command after selecting a resource file, depending on the context.

- If you are the creator of the file, then you see the **Delete** command, and you can delete your local copy of the file. (You see this command after you create a new class, for example.)
- If the file is part of the base configuration, then Studio does not permit you to delete the file. If you have made a copy of the file, you see **Revert to Base** instead of **Delete**. If you use this option, then Studio deletes your *modified* copy of the file, and, thereafter, it uses the base configuration version of the file.
- If a file exists only in the base configuration, then Studio disables these commands as they do not apply. (You can neither delete the base configuration file nor delete a local copy of the file.)

View the SCM Log

You can view a copy of the SCM log by navigating to the Studio **Tools** menu, then selecting **SCM Console**. Studio opens a popup window that displays the SCM actions as they occur.

View and Edit (More) Files in Studio

With the release of ClaimCenter 5.0, you can manage—meaning view or edit—more Studio resource directly in Studio.

Data Model Extensions → **metadata**. Read-only files that you view *but not edit*:

- Metadata data model files located in `modules\cc\config\metadata` directory (for example, the `dm_cc_*.xml` files)

Other Resources. Files that you can both view and edit:

- `datamappingsplit.xml`
- `datamappingtogether.xml`
- `typecodemapping.xml`
- `typelistmappingsplit.xml`
- `typelistmappingtogether.xml`

New Text Editor

Guidewire has added simple text editor to Studio for use with XML files editable in Studio, but for which no dedicated editor exists. This text editor works with `config.xml` or `extensions.xml`, for example. This text editor is active only if you do not specify an external file editor. Guidewire recommends, however, that you link these types of files to an external editor if you plan on making complicated changes to your files.

New Help Menu PCF Format Reference

With the release of ClaimCenter 5.0, Guidewire exposes the *PCF Format Reference* as a choice on the Guidewire Studio **Help** menu.

Changes in Studio in 5.0

The major changes to Guidewire Studio in ClaimCenter 5.0 include:

- New Development Mode
- Resource View Is Virtual
- Manage Most Resources from Studio
- Library Functions Become Classes and Enhancements
- Studio Options Moved to Tools Menu
- Studio Handles Script Parameters Differently

- Studio Links to SCM System

New Development Mode

Guidewire no longer supports the ability to directly deploy (“hot deploy”) resources from Studio to an application instance on a remote server, even one to which Studio directly connects.

Instead, Guidewire supports two types of application modes or environments:

- Development
- Production

Development mode. In Development mode, you work within Guidewire Studio (typically, in a local file system) and use the Studio editors to modify configuration files and extend the base data model. You must then manually build a .war (or .ear) file and physically move it to the production application server, and redeploy the application. In this way, you can modify your installed configuration and test your changes without impacting a running application server.

Although you cannot push your configuration modifications to a remote server, you can, however, connect to a remote (development) server and debug the server rules, for example. You just cannot modify the rules on the remote server.

Production mode. In Production mode, Guidewire limits certain system functions as a safety precaution so that it is not possible to use the development tools on a production server. For example, Guidewire disables the Internal Tools on a production server.

You set the server mode using the following system parameter:

```
-Dgw.server.mode=(dev|prod)
```

See “Using the Maintenance Run Level” on page 64 in the *System Administration Guide* for more information.

Resource View Is Virtual

With the release of ClaimCenter 5.0, Studio presents a virtual view of the resources available for viewing and editing through the **Resources** tree. Previously, Studio showed application files located in the resources directory in the **Resources** tree. The **Resources** tree displays what appears to be a “flat” file structure, although the files can be physically located in multiple directories. If you edit a file, Studio makes a copy of the file and places it in a special configuration directory. Thereafter, you edit that file, not the original. In the **Resources** tree, Studio indicates edited files by coloring the file name blue.

Manage Most Resources from Studio

With the release of ClaimCenter 5.0, Guidewire provides the ability to manage most ClaimCenter resources from within Studio. Many of these resources have dedicated editors. Other resources are editable in editors external to Studio, but called from Studio. The following resources have dedicated editors:

- Classes
- Display Keys
- Entity Names
- Enhancements
- Page Configuration (PCF)
- Rule Sets
- Tests
- Typelists
- Web Services
- Workflows

Along with the resources with dedicated Studio editors, you can access the following additional resources from Studio by launching an external editor:

- Application configuration (`config.xml`)
- Address Book information
- Currency specification
- Geographical information
- Locale definitions
- Reporting configuration
- Search configuration
- Security configuration
- Static user interface (UI) resources (style sheets, graphics, HTML)

Note: You associate an external editor with a file type in **File** → **Settings** → **External Editors**

Library Functions Become Classes and Enhancements

Studio no longer supports the use of the library functions defined in **Libraries** in the **Resources** tree (which includes both utility and class extension libraries). Instead, these GScript functions become GScript *classes* (for utility library functions) or GScript *enhancements* (for extension library functions).

Library type	New GScript type
Class Extension	GScript Enhancement
Utility	GScript Class

For example, the utility library function **Libraries** → **Utility** → **MyUtil** becomes **Classes** → **libraries** → **Utility** → **MyUtil.gs**. The class extension library function **Libraries** → **Class Extensions** → **Policy** → **MyPolicyExtension** becomes **Classes** → **libraries** → **Policy** → **MyPolicyExtension.gsx**. Notice the reference path for each remains the same. You access **Libraries.MyUtil()** and **Policy.MyPolicyExtension()** in exactly the same way as if they were still library functions. Therefore, this change requires no change to existing GScript code (in PCF pages or business rules, for example) that contains references to the old library functions.

Note: Guidewire provides a convertor utility (as part of the upgrade tool) to handle the conversion of any existing libraries.

Studio Options Moved to Tools Menu

Guidewire has moved the Studio **Options** dialog from the **File** menu to the **Tools** menu. There are many new configuration options available. For more information, see “Configuring Guidewire Studio” on page 91 in the *Configuration Guide*.

Studio Handles Script Parameters Differently

Guidewire has changed how Studio (and the application) handle script parameters. (Script parameters are Studio-managed resource that you can use as “global” variables within GScript code.) On server start-up, ClaimCenter compares the list of script parameters that currently reside in the database to those in the `ScriptParameters` file.

- ClaimCenter adds any script parameters that are in the XML file but not in the database to the database, with whatever initial values are set in the XML file.
- ClaimCenter ignores all other values in the XML file. This means that ClaimCenter explicitly *does not* propagate changes to values in the XML file to the database.

After a script parameter resides in the database, you manage it solely in the **Script Parameters** administration screen from within ClaimCenter itself. You access the **Script Parameters** administration screen by first logging on using an administrative account, then navigating to **Administration** → **Script Parameters**.

Studio Links to SCM System

Releases prior to ClaimCenter 5.0 provided a limited ability to link Studio resource files to a source control management (SCM) system. However, with the release of Guidewire ClaimCenter 5.0, that ability becomes much more robust. In the 5.0 release, Guidewire provides the ability to link your resource files to the following SCM systems:

- CVS
- Perforce
- Subversion (SVN)

Studio provides the following abilities also:

- Ability to connect to a non-supported SCM system using the File Filter Adapter module.
- Ability to manage resource files that do not reside in a SCM system.

If linked to a SCM system, you can do the following in Studio, using the right-click **Source Control** menu commands:

Command	Use to
Add	Mark a newly created file to "add" to the source control system.
Edit	Open (check out) a file for editing. If the file is a base configuration file, Studio makes a copy of the file in the <code>modules/configuration</code> directory. It is this file that you edit, and, it is this file that you must check into source control.
Revert	Revert a file back to its previously checked-in version in source control.
Synch	Synchronize your local copy of the file with the last previously checked-in version of the file.
Submit	Submit your local (modified) version of a file to the SCM system.

Studio and the Source Control System

If you use either CVS or SVN (Subversion) as your SCM system, Guidewire provides a means for you to exclude certain files or directories from the Studio checksum process. You do this through a `scm.regex` statement in the `.../bin/build.properties` file. There are several examples of this statement in the file indicating how to use it, one for CVS and one for SVN.

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.

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

Guidewire ClaimCenter 4.0.0 Release Notes

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

Release 4.0.0.390

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.0.390.

Installation Requirements

For detailed installation requirements, see the *ClaimCenter Installation Guide*.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please email technical support at support@guidewire.com.

Installing This Release

For installation information, please refer to the *ClaimCenter Installation Guide*. For information on installing an upgrade to your existing ClaimCenter installation, refer to the *ClaimCenter Upgrade Guide*.

- If you are running on an Oracle 10.2.0.2 database, you must install Oracle RDBMS patch 4604970.

Changes in this Release

For a description of the changes between ClaimCenter 3 and ClaimCenter 4, refer to the *ClaimCenter Upgrade Guide*.

You should also be aware of the following changes:

- There is no longer a separate installer for Guidewire Studio. Instead, run Studio with the `studio.bat` command in `ClaimCenter/bin`.

Known Issues and Limitations

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

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the "Applying upgrade step ### of ####" log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for Websphere. These can be found in the WSADMIN Console; click Nodes, drill down to your application server, click the Advanced tab, and increase the values.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant *Microsoft Knowledge Base Article 240928*.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** *Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer.* To make this change, create or edit the [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to "yes".

Team page does not display the latest statistics (CC-2911)

Issue: When viewing the Team page in ClaimCenter, if you click Refresh in your browser to show the most recent data, the information displayed does not change.

Workaround: Your web browser may be showing a cached version of the page. To force it to reload the latest information, hold down the Shift key and click Reload.

Large number of recurring checks causes ClaimCenter to stop responding (CC-5731)

Issue: If you create a check with a large number (more than 1000) of recurring payments, ClaimCenter may stop responding.

Workaround: Ensure that your checks contain fewer recurring payments. You can create a transaction validation rule to enforce this.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin configuration in ClaimCenter/config/config.xml.

Financials are incorrect after changing primary coverage on exposure with transactions (CC-6853)

Issue: An exposure has a primary coverage set, and its transactions are based on that coverage. If you change the primary coverage on an exposure after transactions have been created for it, then ClaimCenter's financial calculations—which are based on the exposure's primary coverage—will be incorrect.

Workaround: Do not change the primary coverage on an exposure when transactions exist for that exposure. The ClaimCenter sample rules contain an exposure validation rule that ensures the exposure coverage matches the transaction coverage. You should include this rule in your own system. You can also customize the Exposure data model and make PrimaryCoverage uneditable after it is initially set.

Java JIT compiler options may cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's `-XX:CompileThreshold` option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance; however, the server may become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server may crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

Exception on server startup after enabling JMX (CC-25199)

Issue: If you enable JMX and start the server, you will get the following exception: Error instantiating JSR160 connector

Workaround: In your Tomcat deployment, copy the files `webapps/cc/config/plugins/management/lib/*` into `webapps/cc/WEB-INF/lib`, and then restart the server.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a "plugin interface" called `IApprovalAdapter`. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The `IApprovalAdapter` interface is not supported in ContactCenter.

Exception when using IMessageToolsAPI.acknowledgeMessage() (CC-26024)

Issue: Guidewire is aware of this issue and will address it in a future release.

Guidewire ClaimCenter 4.0.1 Release Notes

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

Release 4.0.1.80

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.1.80.

Installation Requirements

For detailed installation requirements, see the *ClaimCenter Installation Guide*.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please email technical support at support@guidewire.com.

Installing This Release

For installation information, please refer to the *ClaimCenter Installation Guide*. For information on installing an upgrade to your existing ClaimCenter installation, refer to the *ClaimCenter Upgrade Guide*.

- If you are running on an Oracle 10.2.0.2 database, you must install Oracle RDBMS patch 4604970.

Changes in this Release

This section describes the product changes in this release.

- This release now supports the Microsoft Internet Explorer 7 browser.

Configuration Environment Changes

The following are the primary changes to the ClaimCenter configuration environment files, located in ClaimCenter/config:

File	Changes
extensions/base/dm_cc_activity_view.xml	Removed ActivityView.AssignedUserName. Added entity ActivityVacationView. Added: <ul style="list-style-type: none"> • ActivityUnassignedView.AssignedUserName • ActivitySearchView.AssignedUserName • ActivityTeamView.AssignedUserName
extensions/base/dm_cc_claim.xml	Added Claim.AllocatedClaimNumber. Changed Claim.ClaimRpt to nullok="false". Changed ClaimAccess.Permission to createhistogram="true". Added entity AllocatedClaimNumber.
extensions/base/dm_cc_community.xml	Added index Group.groupu1. Added <events/> to UserSettings.
extensions/base/dm_cc_exposure_view.xml	Added ExposureClaimantView.ClaimID.
extensions/base/dm_cc_financials_bulkpay.xml	Added BulkInvoiceDBCheckBuilder to BulkInvoice.
extensions/base/dm_cc_financials_check.xml	Added index Check.checku7.
extensions/base/dm_cc_policy.xml	Added column Retired to index Vehicle.vehicleu1.
extensions/base/dm_cc_workflow.xml	Added subtype entity WorkflowSearchCriteria.
extensions/base/dm_pl_activity.xml	Changed ActivityBase.TargetDate to createhistogram="true". Removed index ActivityBase.activityu2. Added index ActivityBase.activity2.

File	Changes
extensions/base/dm_pl_community.xml	Remove index GroupBase.groupu1. Added <events/> to GroupUser.
extensions/base/dm_pl_contact.xml	Changed the following to exportable="true": <ul style="list-style-type: none"> Address.Latitude Address.Longitude Address.HTMID Address.GeocodeStatus NonPersistentAddress.Latitude NonPersistentAddress.Longitude NonPersistentAddress.HTMID NonPersistentAddress.GeocodeStatus Removed entities: <ul style="list-style-type: none"> ContactWithTravellInfo TravellInfo
extensions/base/dm_pl_upgrader.xml	Added entities: <ul style="list-style-type: none"> UpgradeDBStorageSet UpgradeDBStorageSetColumn UpgradeDBStorageSetResult
extensions/base/dm_pl_workflow.xml	Changed entity WorkflowSearchCriteria to WorkflowSearchCriteria-Base.
extensions/base/dm_pl_workqueue.xml	Added index WorkItem.WorkItemIndex1. Added InstrumentedWorker.LastNotificationTime.
extensions/base/tl_pl_system.xml	Priority typecodes are now internal.
resources/classes/util/document/DocumentProduction.gs	Added new methods to the DocumentProduction class. See CC-25130.
resources/classes/util/Snapshot.gs	Added getViewableNotes() method to Snapshot class.
resources/libraries/Claim/ClaimUI.xml	Added library function areInitialValuesSet().
resources/libraries/Claim/Flagging.xml	Renamed library functions ex_SetFlag() to setFlagExt(), and ex_RemoveFlag() to removeFlagExt().
config.xml	Added configuration parameters: <ul style="list-style-type: none"> ResourcesMutable MetroPropertiesFileName MaxBrowserHistoryItems PrintFontFamilyName PrintFOPUserConfigFile Removed value WC from ShowNewExposureChooseByCoverageTypeMenuForLossTypes parameter.

PCF Syntax Changes

The following are the primary changes to the XML syntax used to define PCF files:

PCF Element	Changes
<AbstractAddMenuItem>	New PCF element available.
<AddMenuItem>	New sub-element available: <ul style="list-style-type: none"> <AbstractAddMenuitem> Sub-element removed and no longer supported: <ul style="list-style-type: none"> <AddMenuitem>
<AddMenuItemIterator>	New PCF element available.

PCF Element	Changes
<AddressBookContactCell>	New attributes available: <ul style="list-style-type: none"> • enableSort • sortBy • sortDirection
<Calendar>	New attribute available: <ul style="list-style-type: none"> • calendarSource
<Card>	New attributes available: <ul style="list-style-type: none"> • onSelect • selectOnEnter
<CategoryCell>	PCF element removed and no longer supported.
<ClaimContactCell>	Attributes removed and no longer supported: <ul style="list-style-type: none"> • action • confirmMessage • fontColor • hasCheckBox • helpText • iconLabel • inputConversion • labelStyleClass • onChange • onPick • outputConversion • postOnChange • requestValidationExpression • setter • showConfirmMessage • styleClass • useHeaderStyle • validationExpression • validationLabel • valueVisible
<ClaimContactInput>	Attribute removed and no longer supported: <ul style="list-style-type: none"> • setter
<ClaimRelatedContactCell>	New PCF element available.
<ClaimTab>	PCF element removed and no longer supported.
<DateCriterionChoiceInput>	Attributes removed and no longer supported: <ul style="list-style-type: none"> • chosenOptionLabel • chosenOptionValueRange
<DetailViewPanel>	New attribute available: <ul style="list-style-type: none"> • compress
<EntityTypeCell>	PCF element removed and no longer supported.
<FinancialCriterionChoiceInput>	Attributes removed and no longer supported: <ul style="list-style-type: none"> • chosenOptionLabel • chosenOptionValueRange
<InfoBar>	New sub-element available: <ul style="list-style-type: none"> • <Code>
<InputIterator>	Attribute autoAdd changed from boolean to string.
<ListDetailPanel>	New attribute available: <ul style="list-style-type: none"> • selectionOnEnter
<MenuItemTree>	New PCF element available.
<NewActivityMenuItem>	PCF element removed and no longer supported.
<NewClaimWizardNewExposureMenuItem>	PCF element removed and no longer supported.
<NewExposureMenuItem>	PCF element removed and no longer supported.

PCF Element	Changes
<Page>	Attribute removed and no longer supported: <ul style="list-style-type: none"> validationType New attribute available: <ul style="list-style-type: none"> handlesValidationIssue
<PanelSet>	New sub-element available: <ul style="list-style-type: none"> <Verbatim>
<RowIterator>	Datatype of attribute autoAdd changed from boolean to string.
<SearchMenuItem>	New attribute available: <ul style="list-style-type: none"> visible
[TextInputType] <i>applies to:</i> <ul style="list-style-type: none"> <BulletPoint> <ConfirmPasswordInput> <TextAreaInput> <TextInput> 	New attribute available: <ul style="list-style-type: none"> inputMask
<ToolbarDownloadButton>	<ul style="list-style-type: none"> New PCF element available.
[WizardStepBase] <i>applies to:</i> <ul style="list-style-type: none"> <QuickClaimStep> <WizardStep> 	Attribute removed and no longer supported: <ul style="list-style-type: none"> validationType New attribute available: <ul style="list-style-type: none"> handlesValidationIssue
<Worksheet>	Attribute removed and no longer supported: <ul style="list-style-type: none"> validationType New attribute available: <ul style="list-style-type: none"> handlesValidationIssue

PCF Implementation Notes

The following are some important notes regarding the default PCF implementations:

PCF Element	Changes
<TabBar>	The Claim tab menu is no longer a custom widget. It is very unlikely you could have changed this, as it wasn't very configurable; but if you did, the custom widget was replaced with a menu item, and the claims are now just a menu iterator.

PCF Element	Changes
<ClaimMenuActions>	<p>The NewActivityMenuItem widget is gone and has been replaced with a reference to the NewActivityMenuItemSet PCF file. The new reference looks like the following:</p> <pre><MenuItem label="displaykey.Web.ClaimMenuActions.NewActivity" shortcut="V" id="ClaimMenuActions_NewActivity"> <MenuItemSetRef def="NewActivityMenuItemSet(Claim)"/> </MenuItem></pre> <p>The NewExposureMenuItem widget is gone and has been replaced with a reference to the NewExposureMenuItemSet PCF file. The new reference looks like the following:</p> <pre><MenuItem label="displaykey.Web.ClaimMenuActions.NewExposure" shortcut="X" available="!Claim.Closed" visible="Claim.ExposureListChangeable" id="ClaimMenuActions_NewExposure"> <MenuItemSetRef def="NewExposureMenuItemSet(Claim)" mode="getNewExposureMenuMode()"/> </MenuItem></pre> <p>along with the following code block:</p> <pre><Code> function getNewExposureMenuMode() : String { var byCoverageType = gw.api.exposure.NewExposureMenuUtil.showChooseByCoverageTypeMenu(Claim); var byCoverage = gw.api.exposure.NewExposureMenuUtil.showChooseByCoverageMenu(Claim); if (byCoverageType and byCoverage) { return "both"; } else if (byCoverageType) { return "coveragetype"; } else if (byCoverage) { return "coverage"; } else { throw new java.lang.IllegalStateException("Claim loss type " + Claim.LossType + " is not configured to show the new exposure menu either by coverage type or by coverage"); } } </Code></pre>

PCF Element	Changes
<NewClaimMenuActions>	<p>The NewClaimWizardExposureMenuItem widget is gone and has been replaced with a reference to the NewClaimNewExposureMenuItemSet PCF file. The new reference looks like the following:</p> <pre><MenuItem label="displaykey.Web.ClaimMenuActions.NewExposure" shortcut="X" visible="Wizard.showNewExposureMenu()" id="NewClaimMenuActions_NewExposure"> <MenuItemSetRef def="NewClaimNewExposureMenuItemSet(Claim, Wizard)" mode="getNewExposureMenuMode()"/> </MenuItem></pre> <p>along with the following code block:</p> <pre><Code> function getNewExposureMenuMode() : String { // Return "none" if there's no claim or no loss type yet if (claim == null claim.LossType == null) { return "none"; } var byCoverageType = gw.api.exposure.NewExposureMenuUtil.showChooseByCoverageTypeMenu(Claim); var byCoverage = gw.api.exposure.NewExposureMenuUtil.showChooseByCoverageMenu(Claim); if (byCoverageType and byCoverage) { return "both"; } else if (byCoverageType) { return "coveragetype"; } else if (byCoverage) { return "coverage"; } else { throw new java.lang.IllegalStateException("Claim loss type " + Claim.LossType + " is not configured to show the new exposure menu either by coverage type or by coverage"); } } </Code></pre>
<AssignActivitiesPopup> <AssignClaimsPopup> <AssignExposuresPopup> <AssignMatterPopup>	Removed the return statement from beforeCommit.
"currentLocation" symbol in TabBar	If you modified out-of-box PCFs and added usages of currentLocation symbol in TabBar or MenuLinks (which are defined for a LocationGroup rather than a concrete Page), please note that the currentLocation has been fixed to point to the top/visible Page, and not the LocationGroup.
<ClaimEvaluations> <ClaimNegotiations>	Added addVisible="false" to <IteratorButtons>, to avoid multiple buttons with a duplicate ID in the same toolbar.
<NewClaimWizard> <ClaimPolicyGeneral>	<p>The validationType attribute is replaced by handlesValidationIssue, to support the dynamic validation issue drilldown.</p> <ul style="list-style-type: none"> old value: validationType="Policy" new value: handlesValidationIssue="VALUE typeis entity.Policy"

Improvements

The following are the primary improvements made to ClaimCenter for this release:

Issue ID	Description
CC-25130	Added new methods to the <code>DocumentProduction</code> class: <ul style="list-style-type: none"> <code>asynchronousDocumentCreationSupported()</code> <code>synchronousDocumentCreationSupported()</code>
CC-25422	Added events to the <code>GroupUser</code> entity. When a user is added to a group, the events generated are <code>GroupChanged</code> (the <code>Users</code> array on <code>Group</code> is considered changed, since <code>Group</code> owns <code>GroupUser</code>) and <code>GroupUserAdded</code> . When a user is taken out of a group, the events generated are <code>GroupChanged</code> and <code>GroupUserRemoved</code> . The <code>UserChanged</code> event is no longer generated in these cases.
CC-25954	Added the new logger category <code>Messaging.Events</code> to show detailed information about what events are getting raised and served to the destinations.
CC-26621	Support for the <code>SystemOfRecord</code> attribute on messaging destinations has been removed from "native" messaging destinations that use ClaimCenter 4.0.0-style messaging plugins due to changes that made it irrelevant in the new messaging architecture (specifically, removing validation levels for destinations). If you use the old-style deprecated message sinks, this attribute continues to work as in previous releases, although Guidewire strongly encourages you to migrate integration code to the new messaging architecture, including the new messaging plugins and event model.
CC-26630	The Email Message Sink has been deprecated. Its functionality has been reimplemented with the newer plugin/destination framework. For more information, refer to the <i>ClaimCenter Integration Guide</i> . The new plugin is enabled in the default <code>config.xml</code> file included with ClaimCenter 4.0.1. If the plugin tag and its corresponding destination are commented out, then the behavior reverts to the deprecated message sink (thus, existing implementations will continue to work until migrated).
CC-26975	<p>There has been a change in the support of the deprecated <code>IDocumentCreationOptions</code> interface. The default <code>ClaimMenuActions.pcf</code> file has a section that looks like this:</p> <pre><MenuAction label="Web.ClaimMenuActions.NewDocument"> <MenuItemSetRef def="ClaimNewDocumentMenuItems(Claim)"/> </MenuAction></pre> <p>In ClaimCenter 4.0.0, to use <code>IDocumentCreationOptions</code>, this code must be the following:</p> <pre><MenuAction label="Web.ClaimMenuActions.NewDocument"> <MenuItemSet id="NewDocumentMenuItems" menuItemSetDef="DocumentCreationOptionsMenuItems" class="DocumentCreationOptionsMenuItems"/> </MenuAction></pre> <p>In ClaimCenter 4.0.1, to use <code>IDocumentCreationOptions</code>, this code must be the following:</p> <pre><MenuAction label="Web.ClaimMenuActions.NewDocument"> <MenuItemSetRef def="DocumentCreationOptionsMenuItemSet(Claim)"/> </MenuAction></pre> <p>This change causes the New Documents menu to be built based on the <code>IDocumentCreationOptions</code> implementation, and when one of those options is selected it causes the option to be executed in a popup window, as before. Note that if your implementation depends on Struts being loaded, or on some other aspect of the ClaimCenter 3.1 implementation, you may need to modify additional PCF code. If any significant work is required, Guidewire recommends that this work be redone using PCF <code><ExitPoint></code> definitions, rather than using this temporary backwards-compatibility mechanism.</p>

General Issues

The following are the primary issues addressed in this release:

Issue ID	Description
CC-25356	Fixed an issue where an erroneous <code>UserChanged</code> event would be generated when visiting a claim that had not been visited before.
CC-27196	Fixed an issue where the <code>IDataObject</code> SOAP API method <code>setPathValues()</code> required a valid root object to do a 'rootless add', even though the root entity passed to the method was not used. This is now fixed so you can pass <code>NULL</code> for the type and id of the root entity to add a root entity.

Known Issues and Limitations

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

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the "Applying upgrade step ### of ####" log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for Websphere. These can be found in the WSADMIN Console; click Nodes, drill down to your application server, click the Advanced tab, and increase the values.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant *Microsoft Knowledge Base Article 240928*.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** *Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer.* To make this change, create or edit the [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to "yes".

Team page does not display the latest statistics (CC-2911)

Issue: When viewing the Team page in ClaimCenter, if you click Refresh in your browser to show the most recent data, the information displayed does not change.

Workaround: Your web browser may be showing a cached version of the page. To force it to reload the latest information, hold down the Shift key and click Reload.

Large number of recurring checks causes ClaimCenter to stop responding (CC-5731)

Issue: If you create a check with a large number (more than 1000) of recurring payments, ClaimCenter may stop responding.

Workaround: Ensure that your checks contain fewer recurring payments. You can create a transaction validation rule to enforce this.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin configuration in `ClaimCenter/config/config.xml`.

Financials are incorrect after changing primary coverage on exposure with transactions (CC-6853)

Issue: An exposure has a primary coverage set, and its transactions are based on that coverage. If you change the primary coverage on an exposure after transactions have been created for it, then ClaimCenter's financial calculations—which are based on the exposure's primary coverage—will be incorrect.

Workaround: Do not change the primary coverage on an exposure when transactions exist for that exposure. The ClaimCenter sample rules contain an exposure validation rule that ensures the exposure coverage matches the transaction coverage. You should include this rule in your own system. You can also customize the Exposure data model and make PrimaryCoverage uneditable after it is initially set.

Java JIT compiler options may cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's `-XX:CompileThreshold` option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance; however, the server may become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server may crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a "plugin interface" called `IApprovalAdapter`. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The `IApprovalAdapter` interface is not supported in ContactCenter.

Contact.Linked() and Contact.Synced() methods are missing (CC-27259)

Issue: These methods were intended to be deprecated, but were mistakenly removed completely.

Workaround: Change:

```
var linked = contact.Linked;
```

to

```
var linkStatusValue = contact.calculateLinkStatus().LinkStatusValue;  
var linked = (linkStatusValue == "NotSynced" or linkStatusValue == "SyncedRemote" or  
             linkStatusValue == "Synced");
```

and change:

```
var synced = contact.Synced;
```

to

```
var linkStatusValue = contact.calculateLinkStatus().LinkStatusValue;  
var synced = (linkStatusValue == "Synced");
```


Guidewire ClaimCenter 4.0.2 Release Notes

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

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.2.84.

Installation Requirements

For detailed installation requirements, see the *ClaimCenter Installation Guide*.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please email technical support at support@guidewire.com.

Installing This Release

For installation information, please refer to the *ClaimCenter Installation Guide*. For information on installing an upgrade to your existing ClaimCenter installation, refer to the *ClaimCenter Upgrade Guide*.

- If you are running on an Oracle 10.2.0.2 database, you must install Oracle RDBMS patch 4604970.

Notes on Upgrading Rules from Release 4.0.x

The upgrade to ClaimCenter 4.0.2 does not automatically merge in changes that you have made to rules in previous 4.0.x releases. Therefore, we recommend that you preserve your customized rules and then, where desired, manually incorporate any changes that Guidewire made to the shipped rules provided with 4.0.2. To do so, use the following procedure:

- Before upgrading, make a backup copy of your entire set of existing rules; that is, the complete ClaimCenter/config/resources/rules directory.
- Run the upgrade tool.
- Copy the rules directory from your backup to the upgraded ClaimCenter/config/resources directory. Allow the copy to overwrite any files that it needs.
- Run Guidewire Studio. Import the rule changes from one (only one!) of the following files supplied with this ClaimCenter release:
 - If upgrading from 4.0.0, import the file: ClaimCenter Rule Diffs 400-402 For Import.xml
 - If upgrading from 4.0.1, import the file: ClaimCenter Rule Diffs 401-402 For Import.xml
- After you import the file, you will see all changed rules appear in Studio, disabled, and under parents named for the date of the import.
- Review the rule differences detailed in the *Base Rule Changes* section of this Release Notes document.
- If you wish to incorporate any of the rule changes into your own rules, do it in Studio. You can either copy the changed section from the imported rule into your rule, or you can delete your rule and replace it with the imported rule. When you are finished, delete any remaining unused imported rules. **WARNING:** You should make the changes only within Studio. Do not attempt to modify the rules XML files directly, or to merge the diffs using a third-party utility.

Changes in this Release

This section describes the product changes in this release.

- “Configuration Environment Changes” on page 206
- “PCF Syntax Changes” on page 208
- “Base PCF File Changes” on page 209
- “Base Rule Changes” on page 209
- “Improvements” on page 209
- “General Issues” on page 211

Configuration Environment Changes

The following are the primary changes to the ClaimCenter configuration environment files, located in ClaimCenter/config:

File	Changes
datadistribution/datadistributions.xml	Added new file.

File	Changes
extensions/base/ dm_cc_aggregateLimit.xml	Changed TmpAggregateLimitRptUpdate.CoverageType to nullok="true".
extensions/base/dm_cc_claim.xml	Changed Claim.LossDate to createHistogram="true". Removed index ClaimContact.claimcontact2. Commented out index ClaimContact.claimcontact4. Added database check UniqueClaimContactDBCheckBuilder. Renamed index UserRoleAssignment.urassign1 to urassignu1. Added index UserRoleAssignment.urassign2.
extensions/base/dm_cc_claim_view.xml	Added new view entity ClaimAssignmentView. Moved the following fields from ClaimAbstractView to ClaimAssignmentView: AssignmentStatus AssignedQueue AssignedUser
extensions/base/dm_cc_claim_view.xml	Added entity ClaimVacationView. Changed entity ClaimTeamView to supertypeEntity="ClaimAssignmentView". Added computed column ClaimTeamView.TotalIncurredNet.
extensions/base/dm_cc_community.xml	Removed index Group.groupu1.
extensions/base/dm_cc_document.xml	Changed the following to type="shorttext": DocumentTemplateSearchCriteria.Identifier DocumentTemplateSearchResults.Lob DocumentTemplateSearchResults.State DocumentTemplateSearchResults.Section DocumentTemplateSearchResults.Identifier
extensions/base/dm_cc_exposure.xml	Changed Exposure.ClaimID to importableagainstexistingobject="false".
extensions/base/dm_cc_financials.xml	Added index Transaction.transaction8. Added index TransactionLineItem.lineitem1. Changed entity TransactionOffsetOnset to overwrittenInStagingTable="true". Added temp entities TmpExposureRptStaging and TmpStagingExposureRpt.
extensions/base/ dm_cc_financials_bulkpay.xml	Changed the following entities to loadable="false": BulkInvoice BulkInvoiceItem ReserveLineWrapper BIVValidationAlert
extensions/base/ dm_cc_financials_check.xml	Added column Status to index Check.checku4. Renamed index CheckPayee.checkpayee1 to checkpayeeu1. Added index CheckPayee.checkpayeeu2. Added temp entity TmpMixedCheckGroups.
extensions/base/dm_cc_note.xml	Added index Note.note6.
extensions/base/dm_cc_policy.xml	Added array column Policy.PeriodPolicies. Removed consistency checks: Policy.PolicyPartiesDBCheckBuilder Policy.PolicyWorkmansCompDBCheckBuilder
extensions/base/dm_pl_activity.xml	Added column Status to index ActivityBase.activity2.

File	Changes
extensions/base/dm_pl_community.xml	Added index <code>UserBase.useru3</code> . Added database check <code>UserBase.UserBaseDBCheckBuilder</code> . Added foreign key column <code>GroupBase.OrganizationID</code> . Added index <code>GroupBase.group1</code> . Added view entity <code>GroupParentView</code> . Changed <code>OrganizationBase.RootGroup</code> to <code>nullok="false"</code> .
extensions/base/dm_pl_contact.xml	Changed entity <code>OfficialID</code> to <code>table="officialid"</code> . Changed entity <code>GeocodeWorkItem</code> to <code>ignoreForEvents="true"</code> .
extensions/base/dm_pl_datadistribution.xml	Added array column <code>TableDataDist.DateSpanDataDists</code> . Added entity <code>DateSpanDataDist</code> . Changed <code>CustomDataDistRequest.Query</code> to <code>type="longtext"</code> .
extensions/base/dm_pl_document.xml	Changed <code>DocumentTemplateSearchCriteriaBase.Keywords</code> to <code>type="shorttext"</code> . Changed <code>DocumentTemplateSearchResultsBase.Name</code> to <code>type="shorttext"</code> .
extensions/base/dm_pl_upgrader.xml	Added entity <code>UpgradeSchemaVersion</code> . Added array column <code>UpgradeInstance.UpgradeDBParameterSets</code> . Added array column <code>UpgradePhase.UpgradeVersionTriggers</code> . Added column <code>UpgradeRowCount.RowCountChange</code> . Added array column <code>UpgradeDBParameterSet.UpgradeDBParameterRows</code> . Added array column <code>UpgradeDBParameterRow.UpgradeDBParameterPairs</code> . Removed columns from <code>UpgradeDBStorageSet</code> : <code>PolicyException</code> <code>IdlePolicyException</code> Added array column <code>UpgradeVersionTrigger.UpgradeVTStatements</code> .
extensions/base/tl_cc_claim.xml	Added typekey <code>DownloadingReport</code> to typelist <code>MetroReportStatus</code> .
extensions/base/tl_cc_document.xml	Added new typelist <code>DocumentType</code> .
extensions/base/tl_pl_accounting.xml	Changed typelist <code>LedgerSide</code> . Set Debit to <code>priority="1"</code> and Credit to <code>priority="2"</code> .
extensions/extensions.xml	Deleted several example entities.
iso/ISO.properties	Added <code>ISO.EventNames</code> property.
typelists/DocumentType.xml	Added typekey <code>ISO match report</code> to typelist <code>DocumentType</code> .
config.xml	Changed <code><destination name="Console Message Logger"></code> to <code>id="68"</code> . Added parameter <code>EnableInternalDebugTools</code> with default value of <code>false</code> . Removed <code>11/22/2007</code> from <code>HolidayList</code> parameter. Added parameter <code>PCFVerificationMode</code> with default value of <code>all</code> .

PCF Syntax Changes

The following are the primary changes to the XML syntax used to define PCF files:

PCF Element	Changes
<code><AbstractAlertBar></code>	New PCF element available.

PCF Element	Changes
[AddActionType]	New attribute available:
<i>applies to:</i>	visible
<AddButton>	
<AddMenuItem>	
<AddressBookContact-Cell>, <ClaimContactCell>, <ClaimRelatedContact-Cell>	New attributes available: action confirmMessage fontColor hasCheckBox helpText iconLabel inputConversion labelStyleClass onChange onPick outputConversion postOnChange requestValidationExpression showConfirmMessage styleClass useHeaderStyle validationExpression validationLabel valueVisible
<MenuIcon>	PCF element removed and no longer supported.
<PickerCell>	New PCF element available.
<Popup>	Attribute removed and no longer supported: autosaveable
<Screen>	New sub-element available: <AbstractAlertBar> Sub-element removed and no longer supported: <AlertBar>
<SearchPanel>	Attribute searchOnEnter changed from boolean to string.

Base PCF File Changes

To view a report of the changes in the base PCF files, refer to the `readme_files` directory on your local disk.

Base Rule Changes

To view a report of the changes in the base rules from ClaimCenter release 4.0.0 to 4.0.2, refer to the `readme_files` directory on your local disk.

To view a report of the changes in the base rules from ClaimCenter release 4.0.1 to 4.0.2, refer to the `readme_files` directory on your local disk.

Improvements

The following are the primary improvements made to ClaimCenter for this release:

Issue ID	Description
API Issues	

Issue ID	Description
CC-15800	The PolicyDataUtil API--previously available in ClaimCenter 3.1 but then removed from ClaimCenter 4.0--has been restored.
CC-26855	The exception handling on calls to the IAddressBookAdapter now includes the message from the caught exception in the message displayed to the user.
CC-28143	<p>There have been several updates to Metro report integration.</p> <p>There are new event messages:</p> <p>Download Document For HasReport: This event is fired when we get the 'report is ready' notification from Metropolitan.</p> <p>Download Document For Closed: This event is fired when we get the 'report is closed' notification from Metropolitan.</p> <p>There are new functions in the Metro GScript library that allow you to customize the document entity before sending it to the document plugin:</p> <p>downloadHasReportDocument(metroReport : MetroReport): This is called when the Download Document For HasReport event is fired.</p> <p>downloadClosedDocument(metroReport : MetroReport): This is called when the Download Document For Closed event is fired.</p> <p>In the MetroReport entity, there are additional scriptable functions:</p> <p>downloadDocument()</p> <p>createMetroDocument()</p>
CC-28285	Exceptions thrown within the IDocumentContentSource plugin now are surfaced to the user.
CC-28674	IClaimAPI.migrateClaim() now sets the claim state to "open" by default. A new version of this method is provided that takes a ClaimState parameter so you can have the claim set to a different state instead.
GScript Issues	
CC-28185	Added method gw.api.StringUtil.displayElided(), which allows you to specify a maximum string length to be displayed, with additional characters represented by ellipses.
CC-28788	Added new Claim.createCheck() method.
CC-28804	Exposure.createCheck() now optionally takes a paymentType parameter, which specifies the type of payment (for example, final, partial, and so on).
CC-28905	The Retired attribute is now available as one of the properties exposed on the TypeKey entity in GScript.
PCF Issues	
CC-15485	You can now display open reserves in claim search results using ClaimSearchView.OpenReserves.
Server Issues	
CC-26742	The env attribute can now be set on the following configuration parameters: ReportProxyScheme, ReportProxyServerPort, ReportProxyURI, ReportManagerURL, ReportAdminLogin.
CC-28018	Temporary copies of documents created from a template are now cleared from the temp directory. This occurs when the user logs in, when the browser is refreshed, and when a local edit is discarded.
CC-28205	The ant task build-war now empties its tmp directory before running.
CC-28238	The fields on DocumentTemplateSearchResults have been changed to type shorttext.
CC-28358	The examples have been removed from the extensions.xml file.
CC-28579	The ant scripts in ClaimCenter/bin/eclipse have been updated.
CC-28719	The regen-pcfmapping tool now escapes quotes within the data.
CC-28829	The Oracle JDBC driver has been updated to version 10.2.0.2.
CC-29233	<p>The excluded parties and covered parties of a policy are now allowed to share contacts.</p> <p>The policy insured is now allowed to be in the excluded parties or covered parties lists.</p>

Issue ID	Description
CC-29844	Claim pre-update and validation rules no longer fire solely because of events. In other words, Claim pre-update and validation rules run only when actual entity data has changed, and in cases where events are fired that do NOT correspond to already-changed entities, the event firing alone will not trigger Claim pre-update and validation rules. This does not affect most events, since most events correspond to actual entity data changes. However, for the ResyncClaim event (triggered from a claim Resync from the user interface), no entity data is inherently changed from this event, so this change will affect Resync handling. This change will also affect any other custom event firing through the addEvent entity method. NOTE: For customers using ContactCenter, this change also affects ContactCenter validation for events such as the ResyncABContact event.

General Issues

The following are the primary issues addressed in this release:

Issue ID	Description
API Issues	
CC-21688	Fixed an issue where a <none selected> in an extension array would be passed in via a plugin as a NULL entity.
CC-29051	Fixed an issue where IDocumentContentSource.updateDocument() would not save changes to entities related to the Document unless the Document itself was also changed.
CC-28979	Fixed an issue where an internal SQL statement containing empty parentheses would cause a database error. To avoid this, IClaimFinancialsAPI.addClaimFinancialsWithValidation() must now specify at least one payee.
CC-29170	Fixed an issue where IContactAPI.submitUpdates() would not allow setting the PublicID.
GScript Issues	
CC-26987	Fixed a GScript issue where you could not use static variables from other static variables.
CC-28206	Fixed an issue where ISOClaimSearchRequest.createClaimsDriverInfo() would attach the drivers license information element to the ClaimsParty Aggregate rather than inside of the ClaimsDriverInfo Aggregate.
CC-28277	Fixed an issue where a GScript class extension library with the same name as another one would prevent viewing of the first one.
CC-29095	Fixed an issue where Studio would not respond when saving GScript classes containing references to pcf.ClaimLossDetails.
Server Issues	
CC-25899	Fixed an error where ClaimCenter would consider the same configuration parameter defined with multiple different environments to be duplicates.
CC-27776	Fixed an error that would occur when using import_tools to import non-ASCII characters.
CC-28184	Fixed an issue where the mode of snapshot pages depended on the typecode name of the loss type, rather than the typecode code.
CC-28235	Fixed an error where a document template could not be found if the directory name contained non-ASCII characters.
CC-28375	Fixed an error with template_tools.
CC-28474	Fixed an issue where the env attribute for the EnableISOMessageSink and ISOPropertiesFileName configuration parameters could no longer be set.
CC-28609	Fixed an issue with the regen-pcfmapping tool, where the MasterMappingFile.csv file was creating unuable values in the column "Display Name".
CC-28667	Fixed an issue where export_tools ran out of memory.
CC-28709	Fixed an issue where a change in the AssignmentStatus typelist caused the upgrade to fail.
CC-28746	Fixed an issue where the presence of certain database statistics would cause an upgrade from ClaimCenter 2.1.5 to fail.
CC-28881	Fixed an issue for claims sent to ISO, where the ClaimsPayment aggregate was located in the wrong place in the data object.
CC-29106	Fixed an issue where the database upgrader on an Oracle database would cause an ORA-00936 error.

Issue ID	Description
CC-29231	Fixed an issue where the ISO message sink would not respond to the ClaimResync event.
CC-29438	Fixed an issue where a message in retryable error state could not be retried.
User Interface Issues	
CC-27001	Fixed an error that would occur when editing a check, removing the last payment, adding it back, and then trying to save the transaction.
CC-27276	The misnamed ContactRole typekey "Legal Paralegal" has been corrected to "Lead Paralegal".
CC-27403	Fixed an issue where you could not type a claim number in the Claim tab menu if that claim number contained a character already highlighted in the menu as a shortcut key.
CC-28020	Fixed an error that would occur when clicking the link of a property in the PropertiesLV.
CC-28075	Fixed a JavaScript error that would occur if ClaimCenter was launched from another browser window.
CC-28200	Fixed an issue where reserve amounts are incorrect if you take a check that initially exceeds reserves and change its amount so that it no longer does.
CC-28348	Fixed an error when pushing to an ExitPoint from a Popup pushed to from a Worksheet.
CC-28529	Fixed an issue where choosing to print a claim would leave the cursor an hourglass.
CC-28575	Fixed an error that occurred when uploading a document named with non-ASCII characters. If running on Tomcat, note that the Tomcat server requires the proper configuration. See CC-28575.
CC-28576	A vehicle or property that had no coverages would be displayed in the New Exposure menu.
CC-28635	Fixed an issue where the page would not refresh after returning from an Address Book search.
CC-28742	The requestValidationExpression attribute on a ConfirmPasswordInput widget was being ignored.
CC-28753	A claim could not be saved in the new claim wizard if the claimant on an unsaved exposure was changed.
CC-28762	Fixed an issue where upgraded admin data would not appear in the user/group tree.
CC-28889	Transferring a check from one claim to another would cause an erroneous validation error.
CC-28934	Vehicle field would not be cleared when creating one document, then creating another.
CC-28974	Fixed error caused by trying to create an exposure for a vehicle with no coverages in the new claim wizard
CC-29016	Fixed an issue where the profiler internal tool could still be accessed with the keyboard shortcut when disabled.
CC-29096	Fixed an issue where you could not create a payment if there was a pre-update rule for TransactionSet that changed the TransactionSet such that the TransactionSet validation rule then failed.
CC-29126, CC-29128	Fixed an issue where setting a bulk invoice from a valid claim number to an invalid claim number and back would display incorrect cached reserve line items.
CC-29456	Fixed an issue where a bulk invoice could not change status from Requested or Requesting to Issued.
CC-29900	Fixed an issue where an assignment could not be made if it triggered a validation warning.

Known Issues and Limitations

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

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the "Applying upgrade step ### of ####" log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for Websphere. These can be found in the WSADMIN Console; click Nodes, drill down to your application server, click the Advanced tab, and increase the values.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer.

WARNING *Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to "yes".*

Team page does not display the latest statistics (CC-2911)

Issue: When viewing the Team page in ClaimCenter, if you click Refresh in your browser to show the most recent data, the information displayed does not change.

Workaround: Your web browser may be showing a cached version of the page. To force it to reload the latest information, hold down the Shift key and click Reload.

Large number of recurring checks causes ClaimCenter to stop responding (CC-5731)

Issue: If you create a check with a large number (more than 1000) of recurring payments, ClaimCenter may stop responding.

Workaround: Ensure that your checks contain fewer recurring payments. You can create a transaction validation rule to enforce this.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Java JIT compiler options may cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's `-XX:CompileThreshold` option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance; however, the server may become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server may crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a "plugin interface" called `IApprovalAdapter`. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The `IApprovalAdapter` interface is not supported in ContactCenter.

Contact.Linked() and Contact.Synced() methods are missing (CC-27259)

Issue: These methods were intended to be deprecated, but were mistakenly removed completely.

Workaround: Change:

```
var linked = contact.Linked;
```

to

```
var linkStatusValue = contact.calculateLinkStatus().LinkStatusValue;
var linked = (linkStatusValue == "NotSynced" or linkStatusValue == "SyncedRemote" or linkStatusValue == "Synced");
```

and change:

```
var synced = contact.Synced;
```

to

```
var linkStatusValue = contact.calculateLinkStatus().LinkStatusValue;
var synced = (linkStatusValue == "Synced");
```

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate workitems being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in scheduler-config.xml for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding workitem check-out (the "progressinterval" attribute in the <workqueue> tag).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the System Tools page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Application doesn't respond when uploading a document with non-ASCII characters in its name (CC-28575)

Issue: When running on Tomcat, if you attempt to upload a document with non-ASCII characters in its name, the screen may show an hourglass cursor and not respond. This is Tomcat configuration issue.

Workaround: In the Tomcat/conf/server.xml file, add the following attribute to the <Connector> definition in use: useBodyEncodingForURI="true"

Information about importing voided/stopped payments (CC-16750)

Issue: If you are importing voided/stopped payments on an exposure or claim that is closed, then you need to also be sure to generate necessary offsetting reserve transactions to keep open reserves at zero for the affected exposures/claims. The loader callbacks will auto-generate the necessary offsetting payment for each voided/stopped payment, but unless you add the offsetting reserve transactions to the staging table, you'll end up with open reserves on closed exposures/claims, and this will cause a consistency failure in the database. In addition, you cannot import a new exposure against an existing claim, nor can you import financials against an existing exposure/claim. These issues will be addressed in a future release.

Out of memory errors after reloading configuration many times (CC-28736)

Issue: After reloading the configuration files many times, you may get out of memory errors.

Workaround: Restart the server.

You must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the ab_abaddress table. This will help the database understand the distribution of the addresses and thus improve query performance. To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Cannot load financial data using staging tables (CC-30133)

Issue: Financial data cannot be loaded into ClaimCenter via the staging tables. This affects only customers who plan to convert financial data from legacy systems into ClaimCenter via the staging tables. Guidewire is aware of this problem, and will address it in a future release.

©2002-2007 Guidewire Software, Inc. All Rights Reserved.

Guidewire ClaimCenter 4.0.3 Release Notes

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

Release 4.0.3.127

Copyright © 2001-2007 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.3.127

Installation Requirements

For detailed installation requirements, see the *ClaimCenter Installation Guide*.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please E-mail technical support at support@guidewire.com.

Installing This Release

For installation information, please refer to the *ClaimCenter Installation Guide*.

ClaimContact converts into ISO ClaimParty elements

For customers using the built-in Insurance Services Organization (ISO) support in ClaimCenter, be aware that ClaimCenter 4.0.3 now automatically exports additional contact data to ISO compared to previous ClaimCenter versions. ClaimCenter now sends what Guidewire calls `ClaimContact` entities, which is approximately what ISO calls `ClaimParty` XML elements. Similarly, ClaimCenter maps what Guidewire calls `ClaimContact` roles to what ISO calls a claim party role code (`ClaimPartyRoleCd`). ISO considers `ClaimParty` data non-required but strongly recommended for better claim matching, so ClaimCenter added built-in support for these fields.

The most important part of configuring this conversion is defining mapping codes in your `ISO.properties` file. The new `ISO.properties` file include in ClaimCenter 4.0.3 includes new properties, and you may have to copy them to your local version for this to work properly. There are two new sections to configure. Refer to the “ISO Integration” chapter of the *ClaimCenter Integration Guide* for configuration details of these contact `ISOClaimParty` fields.

New APIs for Improved (Editable) XML

ClaimCenter includes new APIs to manipulate XML using XPath path descriptions. You can use some of these new APIs for general GScript use, although some are only for customers implementing ISO integration with custom logic for preparing the ISO payload. In the context of ISO payload generation and also the SOAP API `IDataObject.setPathValues`, XPath support was available in previous releases. However, the new XPath class permits new features like inspection/reflection into existing XML, copying branches of XML elements to another location, and other features. There are two parts to the improvements in this release:

1. A new class callable from GScript new class to manipulate XML. Access this new class from GScript as `com.guidewire.cc.system.integration.messaging.iso.XPath`, which is implemented as a built-in Java class available from GScript code, but not accessible from Java plugins at this time.
2. Alternative versions of the built-in functions that generate ISO payloads. These new versions functionally do the same thing as the previous versions but return the ISO payload in an `ISO editable payload` object, which contains an editable XPath object in its XPath field. The new alternative versions of the built-in payload generation functions have `editable` in the name after the word `generate`, for example `generateEditablePropertyPayload` instead of `generatePropertyPayload`.

The result of these ISO payload-generation functions contains an XPath field, which contains an instance of the new editable XPath instance of the class `com.guidewire.cc.system.integration.messaging.iso.XPath`. Although the new syntax is slightly more intuitive, the biggest advantage is new methods on the XPath class to get, copy, and remove branches of the XML.

You can still add XPaths at ISO payload generation time as done in previous releases of ClaimCenter, such as the following code:

```
var lossType = isoSink.typeCodeMapper.getAliasByInternalCode("LossCause", "iso",
    exposure.Claim.LossCause)

var xPaths = new String[] {
    "/AdjusterParty/AdjusterPartyInfo/CoverageCd=PROP",
    "/AdjusterParty/AdjusterPartyInfo/LossCauseCd=" + lossType
}
payload = isoSink.generatePropertyPayload(exposure, xPaths)
```

However, starting in ClaimCenter 3.1.5, you can optionally create an editable XML object using the following code:

```
var lossType = isoSink.typeCodeMapper.getAliasByInternalCode("LossCause", "iso",
    exposure.Claim.LossCause)
var editablePayload = isoSink.generateEditablePropertyPayload(exposure, null)
editablePayload.XPath.set("/AdjusterParty/AdjusterPartyInfo/CoverageCd", "PROP")
editablePayload.XPath.set("/AdjusterParty/AdjusterPartyInfo/LossCauseCd", lossType )
// return (get) string payload. If there are errors, log them and return null.
payload = editablePayload.getPayloadLogErrors();
```

For more information about editable payloads and the new XPath class, refer to the “ISO Integration” chapter of the *ClaimCenter Integration Guide*.

ClaimCenter 3.1 Upgrade Note for ISO Editable Payloads

ClaimCenter 3.1.5 introduced similar ISO improvements as described above compared to ClaimCenter 3.1.4. If you are upgrading to ClaimCenter 4.0.3 from ClaimCenter 3.1.5 (or a later version of ClaimCenter 3.1.X), there are slightly differences you must be aware of during upgrade.

Two methods on the `editable payload` object changed names. The method `getPayloadLogErrors()` used to get the string payload from an editable payload is now just `getPayload()`, and is the recommended payload method. The method that used to be called just `getPayload()`, which throws an exception, is now called `getPayloadOrThrow()`. Be sure to update any code that uses these methods, particularly the `getPayload` method, since the behavior of that method name effectively changed between ClaimCenter 3.1.5 and ClaimCenter 4.0.3.

Changes in this Release

This section describes the product changes in this release.

- Configuration Environment Changes
- PCF Syntax Changes
- Base PCF File Changes
- Base Rule Changes
- Improvements
- General Issues

Configuration Environment Changes

The following are the primary changes to the ClaimCenter configuration environment files, located in ClaimCenter/config:

File	Changes
display/base/display.properties	Added, removed, and modified several display keys.
extensions/base/datamodel.xsd	New attribute readonly added to the following element definitions: <ul style="list-style-type: none"> <entity> <subtype>
extensions/base/dm_cc_aggregateLimit.xml	<p>Changed the following properties to type="money":</p> <ul style="list-style-type: none"> TmpAggregateLimitRptUpdate.TotalReserves TmpAggregateLimitRptUpdate.TotalPaid TmpAggregateLimitRptUpdate.TotalPaidNonEroding TmpAggregateLimitRptUpdate.TotalRecoveries TmpAggregateLimitRptUpdate.TotalRecoveryReserves <p>Changed index tmpagglimitupd1:</p> <ul style="list-style-type: none"> Changed attribute unique to false Removed column TransactionID Added column CoverageType
extensions/base/dm_cc_claim.xml	<p>Changed Claim.Activities to exportable="true".</p> <p>Added column ClaimContact.ClaimantFlag.</p> <p>Added dbcheckbuilder</p> <p>Claim.ClaimantFlagDenormDBCheckBuilder.</p>
extensions/base/dm_cc_financials.xml	<p>Changed TransactionSet.ClaimID: removed attribute existingreferencesallowed="false".</p> <p>Changed Transaction.ReserveLineID: added attribute existingreferencesallowed="false".</p> <p>Changed Transaction.ClaimID to existingreferencesallowed="true".</p> <p>Changed entity TransactionOffsetOnset to overwrittenInStagingTable="false".</p> <p>Added index TmpExposureRptStaging.frptupgrade2.</p> <p>Added the following columns:</p> <ul style="list-style-type: none"> TmpStagingExposureRpt.ExposureRptID TmpStagingExposureRpt.ClaimRptID <p>Added the following indexes:</p> <ul style="list-style-type: none"> TmpExposureRptStaging.frptupgrade2 TmpStagingExposureRpt.exposurerpt4 TmpStagingExposureRpt.exposurerpt5
extensions/base/dm_cc_financials_check.xml	<p>Changed CheckPayee.PayeeDenormID to the following:</p> <ul style="list-style-type: none"> loadable="true" overwrittenInStagingTable="true"
extensions/base/dm_pl_availability.xml	<p>New file containing the following entities:</p> <ul style="list-style-type: none"> AvailabilityLookup StateDrivenLookup
extensions/base/dm_pl_community.xml	<p>Changed Credential.UserName to supportsCaseInsensitiveSearch="true"</p> <p>Added index Credential.credentialu2.</p> <p>Changed OrganizationBase.RootGroup to soapnullok="true".</p>
extensions/base/dm_pl_contact.xml	<p>Added index Address.addressu2.</p> <p>Removed the following entities:</p> <ul style="list-style-type: none"> MapImageUrl DrivingDirections DrivingDirectionsElem

File	Changes
extensions/base/dm_pl_staging.xml	Added the following columns: <ul style="list-style-type: none"> LoadInsertSelect.AffectedRowCount LoadMsgSinkInsertSelect.AffectedRowCount LoadCallbackResult.AffectedRowCount LoadCallbackResult.SQLText
extensions/base/dm_pl_upgrader.xml	Added the following columns: <ul style="list-style-type: none"> UpgradeInstance.UpgradePhases UpgradeInstance.UpgradeRowCounts UpgradeInstance.UpgradeTableRegistries UpgradeInstance.UpgradeDBStorageSets UpgradeInstance.UpgradeSteps UpgradeDBStorageSet.UpgradeDBStorageSetColumns UpgradeDBStorageSet.UpgradeDBStorageSetResults Changed the following to nullOk="false": <ul style="list-style-type: none"> UpgradeVTStatement.TableName UpgradeVTStatement.StatementType
extensions/base/dm_pl_workflow.xml	Added foreign key WorkflowSearchCriteriaBase.parent.
extensions/base/tl_cc_exposure.xml	Added typekey ExposureTextType.ISOErrorMessage.
extensions/base/tl_cc_financials.xml	In typelist TransactionLifeCycleState, changed the following: <ul style="list-style-type: none"> typecode new: added categories: <ul style="list-style-type: none"> Payment Reserve Recovery RecoveryReserve typecode draft: added categories: <ul style="list-style-type: none"> Payment Reserve Recovery RecoveryReserve typecode pendingapproval: added categories: <ul style="list-style-type: none"> Payment Reserve Recovery RecoveryReserve typecode rejected: added categories: <ul style="list-style-type: none"> Payment Reserve Recovery RecoveryReserve typecode futuredated: added categories: <ul style="list-style-type: none"> Payment typecode awaitingsubmission: added categories: <ul style="list-style-type: none"> Payment Reserve
extensions/base/tl_cc_history.xml	New file containing typelist HistoryType.
extensions/base/tl_pl_availability.xml	New file containing typelist AvailabilityType.
extensions/base/tl_pl_history.xml	Moved many typekeys to file tl_cc_history.xml.
extensions/base/tl_pl_lookuptablename.xml	New file containing typelist LookupTableName.
config.xml	Added commented-out IMessagingNotification plugin reference.

File	Changes
	<p>Added the following configuration parameters:</p> <ul style="list-style-type: none"> • EntityValidationOrder • Financials • GeocodeWorkerBatchSize • ProximitySearchOrdinalMaxDistance • UnreachableCodeDetection • UpgradeDBParameterPairCacheSize • UpgradeDBParameterRowCacheSize • UpgradeDBParameterSetCacheSize • UpgradeDBStorageSetCacheSize • UpgradeDBStorageSetColumnCacheSize • UpgradeDBStorageSetResultCacheSize • UpgradePhaseCacheSize • UpgradeRowCountCacheSize • UpgradeStepCacheSize • UpgradeTableRegistryCacheSize • UpgradeVTStatementCacheSize • UpgradeVersionTriggerCacheSize <p>Removed the following configuration parameter:</p> <ul style="list-style-type: none"> • DeprecatedEventGeneration

PCF Syntax Changes

The following are the primary changes to the XML syntax used to define PCF files:

PCF Element	Changes
<ButtonInput>	New attribute available: <ul style="list-style-type: none"> • hideIfDisabled
<ConfirmPasswordInput>	New attribute available: <ul style="list-style-type: none"> • passwordValidationExpression
[InputType]	At most only one <Reflect> subelement may appear. If it does, it must now appear before any menu item definitions ([AbstractMenuItem]).
<RadioCell>	New PCF element available.

Base PCF File Changes

To view a report of the changes in the base PCF files, [click here](#) (requires the readme_files directory on your local disk).

Base Rule Changes

The following links require the readme_files directory on your local disk.

- To view a report of the changes in the base rules from ClaimCenter release 4.0.0 to 4.0.3, [click here](#).
- To view a report of the changes in the base rules from ClaimCenter release 4.0.1 to 4.0.3, [click here](#).
- To view a report of the changes in the base rules from ClaimCenter release 4.0.2 to 4.0.3, [click here](#).

Notes on Upgrading Rules from Release 4.0.x

The upgrade to ClaimCenter 4.0.3 does not automatically merge in changes that you have made to rules in previous 4.0.x releases. Therefore, we recommend that you preserve your customized rules and then, where desired, manually incorporate any changes that Guidewire made to the shipped rules provided with 4.0.3. To do so, use the following procedure:

1. Before upgrading, make a backup copy of your entire set of existing rules; that is, the complete ClaimCenter/config/resources/rules directory.
2. Run the upgrade tool.

3. Copy the rules directory from your backup to the upgraded ClaimCenter/config/resources directory. Allow the copy to overwrite any files that it needs.
4. Run Guidewire Studio. Import the rule changes from one (only one!) of the following files supplied with this ClaimCenter release:
 - If upgrading from 4.0.0, import the file: ClaimCenter Rule Diffs 400-403 For Import.xml
 - If upgrading from 4.0.1, import the file: ClaimCenter Rule Diffs 401-403 For Import.xml
 - If upgrading from 4.0.2, import the file: ClaimCenter Rule Diffs 402-403 For Import.xml
 After you import the file, you will see all changed rules appear in Studio, disabled, and under parents named for the date of the import.
5. Review the rule differences detailed in the “Base Rule Changes” section of this Release Notes document.
6. If you wish to incorporate any of the rule changes into your own rules, do it in Studio. You can either copy the changed section from the imported rule into your rule, or you can delete your rule and replace it with the imported rule. When you are finished, delete any remaining unused imported rules.

WARNING You should make the changes only within Studio. Do not attempt to modify the rules XML files directly, or to merge the diffs using a third-party utility.

Improvements

The following are the primary improvements made to ClaimCenter for this release:

Issue ID	Description
CC-25565 and CC-29282	SQL Server 2005 SP2 and SP2a are now supported.
CC-28102	Support for the <title> attribute in Guidewire HTML files now exists.
CC-28443	Recommended fields to ISO integration have been added.
CC-28537	ClaimHistoryHandler now uses display keys, it no longer uses hard coded strings.
CC-28559	HTML title elements are now used in all downloaded HTML reports.
CC-28742	The password requirement of at least 1 alpha and 1 numeric character can now be enforced.
CC-29039	The SQL Server Dynamic Management Views Report now captures Table Scans.
CC-29140	The getActivities() method is now in the SOAP Claim Object; activities are now exportable.
CC-29176	The aggregateLimitCalculations batch process is now internationalized, and it gives an error when the server isn't in maintenance mode.
CC-29181	The SQL Server Dynamic Management Views Report Enhancement now applies AWR style formatting.
CC-29288	DB2 tablespaces are now created with an initial size of 20 MB and grow automatically until reaching 2 GB.
CC-29367	Documents retired from Claim are now removed from the documents linked to an Activity or a TransactionSet.
CC-29427	When changing the Studio source path, Studio completely reloads all resources.
CC-29438	ClaimCenter now has the ability to retry in-flight SOAP messages.
CC-29461	Indexes added to ABAddress to speed up Geocoding search and its related batch job.
CC-29507	Bulk invoices created with its API will not go from Awaiting Submission to Requesting when bulkinvoicesescalation runs.
CC-29513	Activities on the Claim Summary Page no longer show closed activities.
CC-29516	The configuration parameter GeocodeWorkerBatchSize now controls the batch size for queue workers.
CC-29597	You can now add additional validation logic to password entry fields.
CC-29608	A document with a non-ASCII filename can now be seen when the GuidewireDocumentAssistant is disabled.

Issue ID	Description
CC-29620	The ProximitySearchOrdinalMaxDistance configuration parameter now caps the number of results and the maximum distance of the results.
CC-29623	Radio buttons now allow three options in list views, instead of just two.
CC-30195	A delete mechanism has been added to the Upgrade Info page.
CC-30346	The affected row count has been added to the instrumentation of insert/selects in the loader.
CC-30404	A new upgrade trigger now complements 'CheckRpt not retired' when its matching check is retired.
CC-30530	The ISO ClaimSearch XML style sheet version 3.2, provided by ISO, is now part of this release.
CC-30617	Some entities are no longer extensible: To address a previous error, some non-persistent entities are now ordinary Java objects, which means that they are no longer customer-extensible.
CC-30826 and CC-31561	ISO editable payloads and XPaths have been improved. ClaimCenter 4.0.3 now automatically exports additional contact data to ISO. It now sends what Guidewire calls ClaimContact entities, which is approximately what ISO calls ClaimParty XML elements. Similarly, ClaimCenter maps what Guidewire calls 'ClaimContact roles' to what ISO calls a 'claim party role code' (ClaimPartyRoleCd). ISO considers ClaimParty data non-required but strongly recommended for claim matching, so ClaimCenter added built-in support for these fields. See the <i>ClaimCenter Integration Guide's</i> "ISO Integration" chapter for complete details and examples.
CC-31607	There is now ISO Payload Generation Fields reference documentation. Guidewire provides a reference document describing all ClaimCenter fields and their matching fields in the ISO XML. You can find this information in Microsoft Excel format at the path: ClaimCenter/config/iso/ISO_Mapping_Fields.xls
CC-31166	There are new configuration parameters for geocoding. If using Geocoding or Autofill, make sure that these new configuration parameters are set correctly; refer to the Geocoding chapter of the <i>ClaimCenter Application Guide</i> or the <i>ClaimCenter Configuration Guide</i> for details on GeocodeWorkerBatchSize, DrivingDirectionsUseMetricDistances, and ProximitySearchOrdinalMaxDistance
CC-31403	<p>SegmentationResult has Changed from a Java Object to a Non-Persistent Entity. The class returned by ISegmentationAdapter has changed from a Java object to a non-persistent entity. This means that anyone implementing ISegmentationAdapter as a Java plugin will need to change their implementation in the following two ways:</p> <ol style="list-style-type: none"> 1. Class name changes from com.guidewire.pc.plugin.account.SegmentationResult to com.guidewire.cc.external.entity.SegmentationResult 2. To create a new instance of this class, change from new SegmentationResult() to EntityFactoryUtil.newEntity(SegmentationResult.class) <p>In GScript, references to class com.guidewire.cc.plugin.segmentation.SegmentationResult change to entity.SegmentationResult.</p>
CC-31407	<p>IAccountPlugin has changed from a Java object to a non-persistent entity. The class returned by IAccountPlugin.generateAccountNumber has changed from a Java object to a non-persistent entity. This means that if you implement IAccountPlugin as a Java plugin you will need to change the implementation in the following two ways:</p> <ol style="list-style-type: none"> 1. Class name changes from com.guidewire.pc.plugin.account.AccountNumGenResult to com.guidewire.pc.external.entity.AccountNumGenResult 2. To create a new instance of this class, change from new AccountNumGenResult() to EntityFactoryUtil.newEntity(AccountNumGenResult.class) <p>In GScript, references to class com.guidewire.pc.plugin.account.AccountNumGenResult change to entity.AccountNumGenResult.</p>
CC-31112	ClaimCenter again displays financial transactions created by a deleted user.
CC-31942	The ISOReceive servlet now reliably returns an ack to the ClaimCenter messaging system. It used to fail when it found and used a thread carrying insufficient permissions to communicate with the messaging system, which in turn refused to accept the message and waited forever. The ISOReceive servlet now resets the current user on the thread it uses to null; the system then grants the thread superuser permissions, and the messaging system trusts the ack it gets from the servlet.

General Issues

The following are the primary issues corrected in this release:

Issue ID	Description
API Issues (see 'ISO Issues' below for other API Issues)	
CC-23600	No null pointer error occurs when passing an invalid destination id to <code>IMessageToolsAPI.SuspendDestination</code> .
CC-28206	<code>ISOClaimSearchRequest.createClaimsDriverInfo()</code> now generates valid content according to ISO specs.
CC-29743	The <code>CallOperationDTO</code> is now compatible with the <code>telephony.CallOperationDTO</code> .
CC-30657	<code>MessageWriterImpl</code> now calls <code>Transport.suspend()</code> and resumes when the destination is suspended.
CC-30792	The <code>PluginCallbackHandler</code> used by the message reply plugin now bypasses CDCE and validation checks.
CC-31045	Adding a custom event from a messaging plugin no longer increments the bean version or causes an error.
CC-31317	The ack object in the SOAP <code>IMessageToolsAPI</code> exposes <code>AddFieldChanges()</code> and <code>AddCustomEvent()</code> methods.
CC-31531	Implementing the <code>MessageTransport</code> plugin in <code>GScript</code> no longer gives an error.
Financials Issues	
CC-28960	Pay to the Order Of is now populated with all joint payee names, not just the first payee's name.
CC-29042	Check constraints declared on a subtype now work for both consistency checks and integrity checks.
CC-29704	Editing a check recurrence schedule no longer lets you enter a scheduled send date before today.
CC-29804	Changing the policy on a claim now properly updates the <code>AggLimit</code> used.
CC-29870	The <code>paymentType</code> becomes partial - it doesn't stay supplemental - when the onset exposure is open.
CC-30115	Workers' Compensation claims now generate activities and reserves properly when their severity changes.
CC-30295	Before updating a bulk invoice status to cleared, all line items' payments must be in submitting status.
CC-30337	User interface gives no error when you click the Amount link for a Review/Approve Reserve Change activity.
CC-30730	If payee is other than a <code>claimContact</code> , the payto or address is now populated on the Check Payee screen.
CC-30749	An upgrade trigger now handles payments that were recoded and then later voided or stopped.
CC-30790	The <code>reserveLine</code> of an existing payment can now be changed at the same time its eroding status changes.
CC-30949	<code>OffsetOnset</code> loader callbacks don't create offsets/onsets if they're already in <code>ccst_transactionoffsetonset</code> .
CC-31124	No null pointer exception is thrown when creating an Auto Bodily Injury exposure while ISO is enabled.
CC-31414	There is no longer a check for a financials calculation batch process in <code>ClaimAPI</code> , <code>addFNOL</code> and <code>migrateClaim</code> .
CC-31699	Restore New Transaction menu items again use display keys instead of typekey overrides.
CC-31725	<code>AggLimit</code> grouping by <code>Account</code> is not applied on a different policy with the same account.
CC-31770	Each <code>Policy</code> must have the same <code>PolicyType</code> as its <code>PolicyPeriod</code> .
GScript Issues (see also Studio Issues)	
CC-29790	ClaimCenter can now execute a rule or GScript hook when negative acknowledgement for ISO is received.
CC-29855	You can now execute a rule or GScript hook when the ISO message sink is suspended.
CC-30568	The Address Book variant of <code>getContactRelationshipsByRelationship</code> is now exposed and available.
CC-30569	The <code>Contact.Unlink()</code> method remains <code>unlink()</code> .
CC-30606	Staging Import methods no longer allow assignment without an <code>AssignmentStatus</code> .
CC-30671	<code>gw.api.policy.ClaimPolicyMakeEditableUtil.makePolicyEditable()</code> no longer causes an <code>IllegalStateException</code> when <code>ClaimPolicyGeneral</code> is already in Edit mode.
CC-31013	<code>Assignment.assignByUserAttributes()</code> now works as expected; picking the correct group, then a user.
CC-31023	The <code>IDataObjectAPI: getNextSequenceNumber()</code> method no longer returns negative numbers when used with 10 digits.
CC-31164	Exception is no longer thrown for static variables declared in a Gscript class when the class is called as part of processing a reply message.
CC-31308	The <code>getProperties()</code> method of <code>TypeInfo</code> is now thread safe and also has a clear and consistent contract.

Issue ID	Description
CC-31652	The locks CustomLibraryManager and TypeLoadAccess no longer cause deadlocks with each other.
Internationalization Issues	
CC-28989	The Report Menu tree localization can now use apostrophes; this is for internationalization; it works with Cyrillic, too.
CC-29806	PolicyTypes now sorts correctly on the New Claim Wizard; this was an internationalization (Cyrillic) issue.
CC-30320	Exporting administrative data containing Cyrillic data now works in both the command line and the Administration user interface.
CC-30396	Opening an administration-exported XML file doesn't give an 'invalid characters' error when it contains Cyrillic.
CC-30972	The date format now contains field validators, so it maintains all date delimiters (for internationalization and Cyrillic data).
ISO Issues (See also API Issues)	
CC-29738	Communication with ISO now avoids this condition: ISOKnown is true but ISOReceiveData is null.
CC-31096	ClaimCenter now processes responses from ISO while its server is in maintenance mode.
CC-31098	ClaimCenter messages to ISO now avoid the 'claimant' type to avoid rejection by ISO's Property LOB.
CC-31125	ISO no longer returns an error when it receives a message with minimum information about a witness.
CC-31126	WorkCompLossInfo (containing EmployeeInfo, a child of ClaimsInvestigationAddRq) no longer causes an ISO exception.
CC-31127	The boat claim information (in the PropertyLossInfo/Watercrafe/ItemDefinition/Model) is now ordered correctly, and there is no ISO exception.
CC-31128	Mobile Vehicle claim info (Model Year) in Auto claim (Mobile home) no longer causes an ISO exception.
CC-31131	ISOReceive now initializes when the first ISO response arrives while ClaimCenter is in maintenance mode.
CC-31132	A 'general liability' Bodily Injury Damage exposure no longer causes an ISO exception.
CC-31366	Contacts with minimum information no longer cause validation errors when sent to ISO. (see also CC-31278)
CC-31676	There is no longer a validateClaimandExposure exception for ISO when the insured's primary address is null.
CC-31696	Error message from ISO of length greater than 255 no longer causes an exception.
PCF Issues	
CC-31345	The TextArea editor no longer automatically changes any special characters.
Reporting Issues, including INetSoft	
CC-25205	INetSoft report printing must be configured for Linux.
CC-30947	INetSoft now reports claims created on the same date as the report properly.
Server Issues, including Database and Batch Process Issues	
CC-28276	The regen-pcfmappings output is now the same as it was in release 4.0.1.
CC-28969	When a Resync Event is triggered in the user interface, Claim pre-update and validation rules run prior to Event rules.
CC-29041	There is no longer a typelist error or server startup error after changing a typelist.
CC-29450	A bean query returning more than 2100 rows no longer causes problems for the SQL server.
CC-29661	DMVInfoDownload and other zero-byte file downloads no longer fail.
CC-29778	There is no duplicate key error in GroupUserAssignmentState during a round robin assignment in a load test.
CC-29860	No null constraint error occurs in AddressBook when adding an Associated Contact to a New Vehicle Incident.
CC-30027	DownloadDMVInfo no longer breaks nondeterministically.
CC-30159	The order in which entities are validated is no longer hard-coded, but is now configurable.
CC-30545	An activityStep using a General Activity now moves the workflow forward.
CC-30571	The sort order of the User Group Tree differs between 3.1.4 and 2.1.5; it now again sorts by name, not ID.
CC-30581	The build war process yields a more informative and not misleading 'deprecated' warning - and no error.
CC-30584	Invalid CSS files are no longer imported or referenced in print_trans.css and print.css.
CC-30607	Illegal break statements are no longer allowed in Studio, so there are no more BranchExceptions at runtime.

Issue ID	Description
CC-30638	Contact-based fields in a popup are now properly saved in the underlying object.
CC-30791	The Sequence utility no longer fails when multiple users submit claims simultaneously.
CC-30793	The <code>installer.xml</code> file now contains <code>classpath</code> references in <code>typedefs</code> .
CC-30927	No null pointer exception appears when searching repeatedly using a 9 digit Zip Code either with or without a street number in the 'line 1' address field.
CC-31025	Downloading the profiler now works, so debug problems in the field are much easier.
CC-31055	Removing a mask from a base field validator no longer causes an exception.
CC-31170	The <code>CheckedValuesToolBarButton</code> reference now refers to the correct file - which is in <code>ExampleCenter</code> .
CC-31192	<code>template_tools.bat</code> now returns more descriptive error messages about its parsing errors.
CC-31425	Running a batch process on a non batch-server no longer gives an error.
CC-31304	ClaimCenter now resets its display name cache when it rolls back a bundle.
CC-31582	ContactCenter now stores the <code>dateTime</code> field correctly.
CC-31695	<code>IllegalStateException</code> doesn't occur on attempting to access a bean of type <code>ClaimAccess</code> with a null bundle.
Studio Issues (See also GScript Issues)	
CC-25090	In Studio, a method's Javadoc now shows up in the Classes display when you press F1.
CC-27216	The JDBC driver in the Guidewire library has been updated.
CC-29747	Users are able to reference static member variables from a class in the <code>ClaimResync</code> event rule.
CC-30365	<code>Exposure.getInitialReserve()</code> no longer throws a null pointer exception when called by Studio rules.
CC-30768	Studio again supports class extensions on object subtypes that reference GScript classes.
CC-30886	The hidden <code>restoreRecordedPlugins()</code> method no longer appears in Studio.
CC-31048	You may now call <code>SequenceUtil</code> in all rule contexts when a transaction is open without receiving an error.
User Interface (See also User Interface Document Issues)	
CC-25234	The Cancel shortcut button now works in both the New Check and Manual Check Wizards.
CC-26952	The Description of the HazardWaste column in <code>cc_claim</code> is now correct.
CC-28435	A user has access to the Vacation tab only when another user is on vacation. If, while a user is inside this page,
CC-30574	all vacationing users return, it allows the first user to exit without an error.
CC-28765	The screen no longer loses focus on list views when using the Add button.
CC-29069	Setting the attribute <code>required=true</code> in <code>CheckBoxInput</code> now forces the user to place a check in the checkbox.
CC-29088	The toolbar Button <code>toolTip</code> attribute can now reference local variables.
CC-29386	No exception occurs when user switches to a second activity queue from the first queue.
CC-29664	When the user interface triggers a resync, <code>MessageContext.CurrentUser()</code> returns the logged-on user, not the 'sys' user.
CC-29821	The <code>FixedPropertyIncident</code> typelist now has the correct <code>SprinklerType</code> description (was <code>Alarm Type for Property</code>).
CC-29848	A 'new activity' star now appears in the user interface when a subrogation assignment activity is created.
CC-29967	You can now add extra fields to the User Search Screen to extend the fields to use to search for users.
CC-30057	The Parties Involved list view is now properly refreshed.
CC-30277	Printing Workers' Compensation claims no longer relies on values in an external typelist.
CC-30520	The Team page displays Cyrillic correctly now.
CC-30570	The Team screen no longer shows negative counts in the Other row.
CC-30573	You can again delete an <code>ABContact</code> in <code>ContactCenter</code> .
CC-30580	If a user is given View script parameters permission, the edit button is no longer displayed on the screen.
CC-30598	The 'overdue' activities filter now works correctly for activities due on the current date, both AM and PM.
CC-30617	After a 'Find User or Group' by proximity search, asking for driving directions no longer gives an error. This involved changing some non-persistent entities, so this will no longer be customer-extensible.

Issue ID	Description
CC-30818	The user interface is changed to allow retry for safe-order message objects, but not non-safe order messages.
CC-31068	The Submit button for Bulk Invoices now clears the Comments field; the user interface also displays them.
CC-31084	A Skip button has been added to the ClaimCenter PCF class maps so that validation tests will not fail.
CC-31085	Printing an entire claim no longer fails.
CC-31278	Contact pickers no longer filter out AddressBook contacts with similar names (like SMITH and Smith).
CC-31493	Claim FNOL snapshots now render properly after upgrading to 4.0.2.
CC-31672	Adding groups to a user no longer causes a blank screen after upgrading to 4.0.2.
CC-31732	All fields of version 3.0.3 snapshot display pages show correctly in 3.0.x snapshots after upgrading to 4.0.
CC-31852	WSS Login and WSS Logout now handle abandoned sessions properly.
CC-31922	Claims and Exposure links on the Team tab do not refresh. In the Team tab, if you view the claims and exposures of a group, then do the same for a different group, the claims and exposures you see will remain those of the first group, and this display will not change until either you login again, or you display a different set of claims/exposures (using other filter criteria), then return to your original view. The claim summary screen and other display screens do not have this problem. This will be fixed in a future release.

User Interface Document and E-mail Issues

CC-29585	When attaching a document, the file name now shows on the Claim/Documents page.
CC-29823	The Document View button now returns documents stored in ImageRight.
CC-29902	Document Templates now contain a correct and reachable reference to an XSD.
CC-29929	The the parameters of <code>Claim.sendEmailWithBody()</code> have been reordered.
CC-30578	Note Templates now are issued with a Note Topic, rather than a blank topic.
CC-29780	There are now better error message when <code>conversionExpression</code> on a <code>PickerInput</code> is the wrong type.
CC-29899	If <code>RestrictSearchesToPermittedItems</code> is false, selecting an unpermitted result takes the user to a better error message.
CC-30440	When <code>addFNOL()</code> fails due to a database violation, it now throws an exception and rolls back the transaction.
CC-30507	The FNOL Mapper now contains detailed error reporting, including an XML response.
CC-31425	A failure in getting driving directions no longer produces a stack trace, but a reasonable error instead.
CC-29777	Error messages due to an object that was never created (like an unallowed transaction) have been improved.
CC-29539	The upgrade suite no longer fails at the start of the second upgrade test class.
CC-29867	Index names for all tables are in sync with prefixes in the 3.1>4.0 table registry during upgrade; Oracle is now ok.
CC-29979	A staging table load now handles importing new transactions against existing claims/exposures.
CC-30017	An exception is no longer thrown when importing an exposure with a null Incident.
CC-30018	A download of Database Upgrade Info no longer fails with a null pointer exception.
CC-30132	Table import now handles null values of the <code>DoesNotErodeReserves</code> column of the transactions staging table.
CC-30631	Upgrade Instrumentation: The collection of row counts and storage information has been made optional.
CC-30859	<code>template_tools.bat</code> now properly upgrades the 2.1 templates to a format compatible for this release.
CC-30977	There are no more Out of Memory Errors when running <code>-export_tools</code> and <code>-import_tools</code> on large databases.

User Interface Wizard Issues

CC-27148	Contact details are now visible in New Claim Wizard, inside Step 4 (Contact Information).
CC-28792	<code>Claim.Claimant</code> on a duplicate claims search now displays properly on page 4 of the New Claim Wizard.
CC-29384	Citation information is now retained when the New Claim Wizard claim is saved.
CC-30110	There is no error when changing to a new policy in the New Claim Wizard, then canceling the Wizard.
CC-30445	<code>ViewContactDetail</code> is always available in the New Check Wizard; not just after a user creates a new contact.
CC-30579	The New Claim Wizard now displays coverages added later properly in its Exposure Creation menu screen.
CC-30647	<code>getPrevPaidOrScheduledServicePeriod()</code> in the New Check Wizard no longer assumes the contact is already in the database.

Issue ID	Description
CC-31287	Assigning an invalid user in the New Claim Wizard now gives a better error and returns properly to the Wizard.

Known Issues and Limitations

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

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the `[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess]` registry entry. Set the data type to `REG_SZ`, and then set the data value to `yes`.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Java JIT compiler options may cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's `-XX:CompileThreshold` option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance; however, the server may become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server may crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a plugin interface called `IApprovalAdapter`. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The `IApprovalAdapter` interface is not supported in ContactCenter.

Contact.Linked() and Contact.Synced() methods are missing (CC-27259)

Issue: These methods were intended to be deprecated, but were mistakenly removed completely.

Workaround: Change:

```
var linked = contact.Linked;
```

to

```
var linkStatusValue = contact.calculateLinkStatus().LinkStatusValue;
var linked = (linkStatusValue == "NotSynced" or linkStatusValue == "SyncedRemote" or
    linkStatusValue == "Synced");
```

and change:

```
var synced = contact.Synced;
```

to

```
var linkStatusValue = contact.calculateLinkStatus().LinkStatusValue;
var synced = (linkStatusValue == "Synced");
```

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or

- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **Server Tools** page (alt-shift-T from the **Administration** page), or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Application doesn't respond when uploading a document with non-ASCII characters in its name (CC-28575)

Issue: When running on Tomcat, if you attempt to upload a document with non-ASCII characters in its name, the screen may show an hourglass cursor and not respond. This is Tomcat configuration issue.

Workaround: In the `Tomcat/conf/server.xml` file, add the following attribute to the `<Connector>` definition in use: `useBodyEncodingForURI="true"`

Information about importing voided/stopped payments (CC-16750)

Issue: If you are importing voided/stopped payments on an exposure or claim that is closed, then you need to also be sure to generate necessary offsetting reserve transactions to keep open reserves at zero for the affected exposures/claims. The loader callbacks will auto-generate the necessary offsetting payment for each voided/stopped payment, but unless you add the offsetting reserve transactions to the staging table, you'll end up with open reserves on closed exposures/claims, and this will cause a consistency failure in the database. In addition, you cannot import a new exposure against an existing claim, nor can you import financials against an existing exposure/claim. These issues will be addressed in a future release.

You must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance. To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in `filename`.

Out of memory errors after reloading configuration many times (CC-28736)

Issue: After reloading the configuration files many times (perhaps 10-12 times), you may get out of memory errors. This was a problem in 4.0.2 as well; it will be addressed in a future release.

Workaround: Restart the server.

Update to Claim Loss gives an error (CC-30754)

Issue: If you search for a claim without an association, then go to its **Loss Details** screen and add, then update, a claim association, you will receive the error "The Claim you are trying to update was changed by another user. Please cancel and retry your change."

Workaround: Make autosave locations false; this has the drawback that intermediate changes won't be saved when one moves between pages.

Bulk Invoices created with Bulk Invoice API will not go from Awaiting Submission to Requesting when bulkinvoicesescalation is executed (CC-31567)

Issue: ClaimCenter 4.0.2 and earlier had the incorrect behavior that the bulkinvoicesescalation batch process did not escalate API-created bulk invoices; version 4.0.3 corrects this. Still there is an issue of the time of the invoice; if late enough on the escalation date, it might be ignored.

Workaround: the ScheduledSendDate should carry a zero time value.

Upgrading GScript resources overwrites all Classes and Libraries, but not Rules (CC-31908)

Issue: There is a big difference between importing rules into Studio/resources vs. importing classes and libraries, namely that when you import classes and libraries, you overwrite what's already there. To prevent loss of your custom changes and additions to Gscript classes and libraries additions, Guidewire no longer supplies libraries and classes export files with the release; the export files for rules are still present in
...ClaimCenter4.0.3\Guidewire\ClaimCenter.

Workaround: To safely import changed and new classes and libraries into 4.0.3, first compare the supplied Araxis merge report (in \Guidewire\ClaimCenter\Readme) with your corresponding files. Then use Studio to transfer (copy and paste) the needed classes and libraries into your own resource files. Be careful to save the old version of your files before updating your GScript classes and libraries. The previous method of importing and merging is still useful for 4.0.3 rules.

Guidewire ClaimCenter 4.0.4 Release Notes

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

Release 4.0.4.124

Copyright © 2001-2007 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.4.124

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact technical support at support@guidewire.com.

Installing This Release

For general installation information, please refer to the *ClaimCenter Installation Guide*. The following issues pertain to installing this release in particular:

An AssignmentStatus Column Change May Require Upgrading

The AssignmentStatus column of all assignable entities was nullable in releases 4.0.0 through 4.0.3; it is non-nullable in 4.0.4. If you are upgrading from 4.0.0 through 4.0.3, it will take a considerable amount of time for the database upgrade. This is not an issue if upgrading from 3.1.x; this column is also non-nullable in these versions.

A New Plugin Method is Required

Starting in ClaimCenter 4.0.1, there is a new IClaimNumGenAdapter plugin method called `cancelNewClaimNumber()`. You must implement the method, but it is acceptable to have that method do nothing. This method gives plugin you a chance to cancel a previously allocated claim number. It is called in the rare case that the `generateNewClaimNumber()` method was called to allocate a claim number but the claim fails validation and is subsequently canceled or otherwise unfinished. In this case, ClaimCenter calls the `cancelNewClaimNumber()` method to give the claim number generator chance to reuse the previously-allocated number. It takes a claim number and also a customer-defined template data string (generated from a GScript plugin template, as with the other methods in IClaimNumGenAdapter). **WARNING:** Even if you implement `cancelNewClaimNumber()` and perform some intelligent “reacquiring” of the number, there is still a small chance of losing a claim number. For example, if you have a power failure after allocating the claim number but before the change is committed to the database.

You Must Manually Upgrade the Rule Sets

To provide the new Transaction Pre-setup rule set (see CC-30355), you must manually copy a file:

1. Create the following new directory in your existing ClaimCenter installation: `ClaimCenter/config/resources/rules/PR/Transaction Pre-setup`.
2. With this ClaimCenter upgrade package, in the same directory as these release notes, locate the file `ruleset.xml`.
3. Copy the `ruleset.xml` file into the new Transaction Pre-setup directory you created in step 1.

Acknowledgement Changes

Although the `AckHandler` class is deprecated in ClaimCenter 4.0.x, the SOAP entity `Acknowledgement` is not deprecated from the SOAP API because it's still used with the SOAP API interface `IMessagingAPI` method `submitAcknowledgement()`, which continues to exist in ClaimCenter 4.0.x. That API continues to be valid as a method to acknowledge a message directly from an external system, although it's not as flexible as using the new messaging plugins, which permit field updates during acknowledgement (or field updates during errors), or other custom actions.

However, as of ClaimCenter 4.0.1, the SOAP entity subclasses of `Acknowledgement`—specifically `ErrorAcknowledgement` and `DuplicateAcknowledgement`—are now deprecated. In previous releases, these subclasses of `Acknowledgement` identified errors and duplicates when used with the SOAP APIs. Starting in ClaimCenter 4.0.1, you should appropriately set some new fields on the `Acknowledgement` SOAP entity as follows to indicate an error or duplicate:

- Set the `Acknowledgement.error` field to `true` if there was an error.
- Set the `Acknowledgement.duplicate` field to `true` if the message was detected as a duplicate message. Also set `Acknowledgement.error` to `true`.
- Set the `Acknowledgement.retryable` field to `true` if the error is retryable. Note that in this case, you should also set `Acknowledgement.error` to `true`.

Note that a message could be retryable if the error at the destination system is temporary (for instance, if the record to be updated is locked) and there is no mechanism in the integration code for automatic retries. If the message is marked with a retry error, then a ClaimCenter user can request a retry through the user interface.

Also note that there are other ways to submit acknowledgements directly from Java messaging plugins using methods directly on the `Message` entity; see the *ClaimCenter Integration Guide* in the “Events and Messaging” chapter for details.

On a related note, starting in ClaimCenter 4.0.x, all errors thrown during the original messaging plugin `Send()` method are assumed to be retryable. In other words, there is no such thing as a non-retryable exception thrown during the `Send()` process. However, submitting an Ack or error using the SOAP API and the `Acknowledgement` SOAP entity still permits you to submit retryable errors and non-retryable errors, and they will be handled differently in ClaimCenter.

Changes in this Release

This section describes the product changes in this release.

- Configuration Environment Changes
- PCF Syntax Changes
- Base PCF File Changes
- Base Resource Changes
- Improvements
- General Issues

Configuration Environment Changes

The following are the primary changes to the ClaimCenter configuration environment files, located in `ClaimCenter/config`:

File	Changes
<code>display/base/display.properties</code>	Added and modified several display keys.

File	Changes
extensions/base/datamodel.xsd	New attribute showRetiredBeans added to the <viewEntity> element definition.
extensions/base/dm_cc_activity.xml	Added dbcheckbuilder Activity.ActivityDBCheckBuilder.
extensions/base/dm_cc_claim.xml	Changed array Claim.ClaimsAssociatedWith: <ul style="list-style-type: none"> removed attribute owner="true" added attribute triggersValidation="true" Added ClaimantFlag to index ClaimContact.claimcontact1. Added column MetroReport.MetroProcessID.
extensions/base/dm_cc_exposure.xml	Added dbcheckbuilder Exposure.VerifyConsistencyAmongInjuryRelatedFieldsOfExposureEntity. Added index Exposure.exposureu3.
extensions/base/dm_cc_financials_recovery.xml	Changed Recovery.OffsettingRecoveryReserveID to loadable="true".
extensions/base/dm_cc_financials_taccount.xml	Changed TAccount.Entries to owner="true".
extensions/base/dm_cc_financials_view.xml	Changed entity TransactionSearchView to showRetiredBeans="false". Changed entity TransactionView to showRetiredBeans="false".
extensions/base/dm_cc_messaging.xml	Changed Message.ClaimID to remove attribute setterScriptability="hidden".
extensions/base/dm_cc_segmentation.xml	New file containing the following entity: <ul style="list-style-type: none"> SegmentationResult
extensions/base/dm_pl_activity.xml	Removed dbcheckbuilder ActivityBase.ActivityDBCheckBuilder.
extensions/base/dm_pl_assignment.xml	Changed Assignable.AssignmentStatus to nullok="false".
extensions/base/dm_pl_batchprocessing.xml	Added subtype InstrumentedJavaTask.
extensions/base/dm_pl_contactcenter.xml	Added array UpdateBatch.ValidationWarningsToIgnore. Added entity ValidationWarning.
extensions/base/dm_pl_note.xml	Renamed the following NoteTemplateSearchCriteriaBase field: <ul style="list-style-type: none"> DocumentType to NoteType Renamed the following NoteTemplateSearchResultsBase field: <ul style="list-style-type: none"> DocumentType to NoteType
extensions/base/tl_pl_activity.xml	Removed typekey ActivityType.policyapproval.
typelists/ContactSearchType.xml	Added the following typekeys: <ul style="list-style-type: none"> ContactSearchType.externalA ContactSearchType.externalB
typelists/NoteType.xml	Removed typekey NoteType.General. Added typekey NoteType.Diagram.
config.xml	DefaultCacheSize parameter initial value changed to 2000. ClaimCacheSize parameter initial value changed to 2000. New parameters added: <ul style="list-style-type: none"> DisableIndexFastFullScanForClaimSearch DisableIndexFastFullScanForRecoverySearch Parameter removed and no longer available: <ul style="list-style-type: none"> DeprecatedEventFiltering

File	Changes
	<p>New configuration parameters available (but not included in the file by default):</p> <p>PrintClaimNoteBlockSize—When printing a claim with many notes, the notes will be split into blocks of this size, with a title page introducing each block of notes. Large block sizes consume more memory during printing; for example a block size of several thousand could cause an out of memory error.</p> <p>PrintMaxPDFInputFileSize—When printing to PDF, ClaimCenter first creates an intermediate XML file as input to a PDF generator. If the input is very large, the PDF generator may run out of memory. Setting this parameter to a non negative value caps the size of the XML input file; if you attempt to print something that violates this cap, you get an error.</p>

PCF Syntax Changes

The following are the primary changes to the XML syntax used to define PCF files:

PCF Element	Changes
<ArrayDiffRow> <Differences> <FieldDiffRow>	PCF elements removed and no longer available.

Base PCF File Changes

To view a report of the changes in the base PCF files, [click here](#) (requires the `readme_files` directory on your local disk).

Base Resource Changes

The following links require the `readme_files` directory on your local disk.

- To view a report of the changes in the base resources from ClaimCenter release 4.0.0 to 4.0.4, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.1 to 4.0.4, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.2 to 4.0.4, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.3 to 4.0.4, [click here](#).

Notes on Upgrading Rules from Release 4.0.x

The upgrade to ClaimCenter 4.0.4 does not automatically merge in the changes that you have made to rules in previous 4.0.x releases. Therefore, we recommend that you preserve your customized rules and then, where desired, manually incorporate any changes that Guidewire made to the shipped rules provided with this release. To do so, use the following procedure:

1. Before upgrading, make a backup copy of your entire set of existing rules; that is, the complete `ClaimCenter/config/resources/rules` directory.
2. Run the upgrade tool.
3. Copy the `rules` directory from your backup to the upgraded `ClaimCenter/config/resources` directory. Allow the copy to overwrite any files that it needs.
4. Run Guidewire Studio. Import the rule changes from one (only one!) of the following files supplied with this ClaimCenter release:
 - If upgrading from 4.0.0, import the file: `ClaimCenter Rule Diffs 400-404 For Import.xml`

- If upgrading from 4.0.1, import the file: ClaimCenter Rule Diffs 401-404 For Import.xml
- If upgrading from 4.0.2, import the file: ClaimCenter Rule Diffs 402-404 For Import.xml
- If upgrading from 4.0.3, import the file: ClaimCenter Rule Diffs 403-404 For Import.xml

After you import the file, you will see all changed rules appear in Studio, disabled, and under parents named for the date of the import.

5. Review the rule differences detailed in the “Notes on Upgrading Rules from Release 4.0.x” section of this Release Notes document.
6. If you wish to incorporate any of the rule changes into your own rules, do this in Studio. You may either copy the changed section from the imported rule into your rule, or you may delete your rule and replace it with the imported rule. When you are finished, delete any remaining unused imported rules.

WARNING You should make the changes only within Studio. Do not attempt to modify the rules XML files directly, or to merge the diffs using a third-party utility.

Improvements

The following are the primary improvements made to ClaimCenter for this release:

Issue ID	Description
CC-33556	The DefaultCacheSize parameter has been increased in config.xml from 500 to 2000. This is a safe change that can avoid problems in your implementation
CC-31403	The class returned by ISegmentationAdapter has changed from a Java object to a non-persistent entity. If using ISegmentationAdapter, make these two changes: 1. Change the class name from com.guidewire.cc.plugin.segmentation.SegmentationResult to com.guidewire.cc.external.entity.SegmentationResult 2. To create a new instance of this class, change from newSegmentationResult() to EntityFactoryUtil.newEntity(SegmentationResult.class) In GScript, references to class com.guidewire.cc.plugin.segmentation.SegmentationResult change to entity.SegmentationResult.
CC-31401	Several XPath API calls have been renamed. For example, igetPayloadLogErrorsding() has become getPayload(), and getPayload() is now called getPayloadOrThrow().
CC-31399	You may now create checks with more than two check payees.
CC-30355	A new Transaction Pre-setup rule set is available. This allows ClaimCenter to provide better support for modifying the summary amounts of already existing single and recurring checks from rules and in the user interface with regard to adding new payment line items. Here is the primary use case: After a payment is submitted, you should be able to add a new payment to any check in “Awaiting Submission” status, both from the user interface and programmatically. For example, you may wish to add a line item for a certain type of tax computed in rules or to adjust a future payment. This change causes the T-Accounts to update properly. You may now: 1) Add to or remove a payment from a check 2) Add to or remove a line item from a payment 3) Modify a line item's amount
CC-31166	New config.xml parameters have been added to customize Geocoding: GeocodeWorkerBatchSize, DrivingDirectionsUseMetricDistances, and ProximitySearchOrdinalMaxDistance. The <i>ClaimCenter Integration Guide</i> has details.
CC-30942	Some message sink IDs are reserved and cannot be used. For integration with external systems, the valid range for customer destination IDs is 0 through 63, inclusive. All other destination IDs are reserved by Guidewire for built-in destinations.
CC-27266	The geocoding writer daemon is now implemented for both ContactCenter and ClaimCenter.
Database Improvements	

Issue ID	Description
CC-33612	ClaimCenter works around Oracle bug 5886252 by disabling index fast full scan when executing certain searches in Oracle databases. The disabling is controlled by two new <code>config.xml</code> parameters, <code>DisableIndexFastFullScanForClaimSearch</code> and <code>DisableIndexFastFullScanForRecoverySearch</code> . They are both set <code>true</code> by default, but you may disable the workaround by setting them <code>false</code> . If a future version of Oracle fixes the defect, these parameters may be removed. The parameters have no effect on databases other than Oracle.
CC-33426	Optimizer tracing support for Oracle has been added to the session profiler.

General Issues

The following are the primary issues corrected in this release:

Issue ID	Description
Financial Issues	
CC-33343	The <code>VoidStopOffset</code> loader callback is now idempotent; it creates a new transaction only if one doesn't already exist.
CC-33339	The loader callbacks for populating <code>ExposureRpt</code> and <code>ClaimRpt</code> no longer assume that <code>Rpt ID</code> and <code>Exposure/Claim ID</code> are in sync; this makes data loads run smoothly.
CC-32401	The data loader can load data (claim/exposure/financials, and so on) into both empty and non-empty databases.
CC-33212	Payment creation via the <code>Claim.CreateCheck()</code> method no longer generates a <code>java.lang.RuntimeException</code> when executed from a Claim pre-update rule.
CC-33203	When one or more joint payees are selected for a check created through the Quick Check wizard, the Pay To field on the check is properly updated to included all the selected payee first and last names.
CC-33202	When one or more joint payees are selected for a check created through the New Check wizard and the Manual Check wizard, the Pay To field on the check is properly updated to included all the selected payee names.
CC-32985	The "Exposure Only" filter on the financials pages now filters out all Claim-level costs.
CC-32492	The <code>Exposure.setAvailableReserves()</code> method no longer fails when called from the Initial Reserves rule set.
CC-32049	When a reserve line is created, then deleted before a commit, it no longer affects T-accounts.
CC-31975	When the financials escalation process handles imported checks, it no longer raises an exception.
CC-31910	The <code>RecToRecReserves</code> entity is now defined as <code>loadable</code> .
CC-31566	
CC-31526	Recoding a final payment from one exposure to another no longer closes the latter when <code>CloseExposureAfterFinalPayment</code> is <code>false</code> .
CC-31436	Fixed an issue relating to a transaction validation error with exposure updates.
GScript Issue (see also Studio Issues)	
CC-32446	<code>Claim.CurrentAssignment.AssignUserByRoundRobin()</code> now returns <code>false</code> when <code>Claim.assignedUser</code> is <code>null</code> .
CC-31664	You can use the new <code>API EntityFactory.addEntity(entity)</code> to add an entity that is read-only or in a different database transaction to the current database transaction. This is particularly helpful when using a "finder" in GScript and you want to add the entity to the current database transaction, so it can be directly changed and also committed with the current database information, or rolled back if there is some reason why the changes in the database transaction should be rolled back.
ISO Integration	
CC-33058	The message contractually required by ISO when a user or rule takes any ISO-related action is now displayed everywhere in the user interface where it is required.
CC-31913	A workers' compensation claim with medical details (or a workers' compensation medical exposure) now triggers claim validation rules when the claim description is filled in, so the exposure can now be sent to ISO automatically.

Issue ID	Description
CC-31870	For ISO integration, if the log messages directory does not yet exist as specified in the <code>ISO.LogMessagesDir</code> field in the file <code>ISO.properties</code> , the directory will be created automatically if possible. This prevents a failure of the server to start when ISO is enabled and the directory does not yet exist.
CC-31373	No <code>ConcurrentDataChangeException</code> occurs when receiving ISO Match Reports and creating documents.
CC-31366	An ISO claimant contact containing only the minimum required fields (just first and last name) to pass validation is no longer rejected.
Other Integration Issues	
CC-31810	The <code>DeprecatedEventFiltering</code> configuration parameter has been removed; it is now always true, and it is needed only if the plugin based messaging architecture isn't used.
CC-33541	You may now set the <code>Claim</code> field on a message. This forces the message to be <i>safe ordered</i> with that claim, even if the original event was not associated with the claim.
CC-33225	For Metropolitan integration, the new Metropolitan <code><ProcessID></code> tag, returned during Order Acknowledgements and Order Inquiries, is now captured and stored until a new message arrives.
CC-31664	In message reply plugins, fixed a <code>ClassCastException</code> thrown when <code>PluginCallbackHandler.add(entityReference)</code> was called in a callback block in some cases.
CC-31316	The <code>getDocumentContentsInfo()</code> method of the <code>IDocumentContentSource</code> plugin is now called just once.
CC-28689	On the administration messaging destination summary page, which shows a summary for claims, the Skip button only skips the blocking message, not all the messages for the claim. The basic functionality has not changed. However, the label and tooltip for the button is now more explicit to reduce confusion with the Skip button on the claim summary page. Also, when you are on the safe-ordered object page, with the list of messages for a claim, and you click several messages to skip, if some of them are in "Pending send" state, previously they might get sent anyway instead of skipped, but now they will get skipped. Also fixed for this page is the fact that even though you skip a pending message, it did not disappear from the page, it stays in the list. This was because of a bug in the way the messages get cached, and the bug has been fixed.
PCF and Typelist Issues	
CC-30898	The <code>NoteType</code> typelist now contains the new diagram typecode.
Server Issues, including Database, Upgrade and Batch Process Issues	
CC-33221	A <code>NullPointerException</code> is no longer thrown during ClaimCenter WebLogic shutdown.
CC-33096	You may now set a validation query for use in test on borrow/return/idle in our connection pool implementation.
CC-33357	Calls from multiple threads to <code>setPropertyValUe()</code> no longer fail.
CC-32839	Various upgrade errors in converting 3.1.4 PCF files to 4.0.x have been corrected.
CC-32838	Session variables no longer cause problems during a build.
CC-32613	When resyncing a claim, the user interface now shows all exceptions that might occur; and you will know if the resync was successful.
CC-32035	<code>LinkStatus</code> generation no longer reports a contact as "Synced" when the contact is out-of-sync.
CC-32480	The upgrader now gives valid SQL when adding a new exposure type.
CC-32402	A bundle refresh no longer causes a query with too many parameters on SQL Server.
CC-32365	<code>NewExposureMenuUtil.getPolicyChecksum()</code> no longer gives a SQL exception "Arithmetic overflow error converting expression to data type int."
CC-32005	Creation of large documents no longer causes the application server to fail.
CC-31941	SOAP calls now clear the current user if they result in an exception.
CC-31006	The regen-datamapping utility no longer switches the values for Precision and Scale in production table representations.
CC-30903	An error in the <code>Metro.fields.Driver.State</code> display key has been fixed.
CC-30283	Fixed issue regarding deadlocks that can occur during an <code>integritycheckandload</code> running against SQL Server.
Studio Issues	
CC-32167	<code>Ru1eset.xml</code> no longer contains duplicate IDs.

Issue ID	Description
CC-31642	FileSystem class is deprecated and Javadoc contains a warning not to use it, even though it appears in the Util package that Studio exposes.
CC-31114	Dragging a rule by placing the cursor on the active/inactive check box no longer deletes the rule.
CC-23600	The internal FileSystem class was never intended to be supported, although it is visible in the developer tool-kit. Customers who inadvertently used the FileSystem.getConfigDir() and getConfigFile() methods found them missing in ClaimCenter 4.0.3. They have been restored in ClaimCenter 4.0.4 temporarily for code that relied on them due to this confusion, but are not supported APIs. Any customer code using this class should change to avoid the FileSystem class as soon as possible. In future releases, this class may not work, or will work very differently, or may be removed altogether. The FileSystem class is not a supported API.

Upgrade Issue

CC-32092	Claim Snapshot HTML pages (located at .../config/pages/templates/snapshot/310) are not converted by the PCF Converter during an upgrade to 4.0.x; they must be converted manually.
----------	--

User Interface (See also User Interface Document Issues)

CC-33349	Sorting a table using a custom field of a foreign key entity works properly.
CC-33329	The financial screen for summary no longer shows menuEnd().
CC-33282	Transaction List View now refreshes after transaction items have been edited.
CC-33208	User name search now uses a parameter for credential name, rather than a literal string.
CC-33164	Incident overview description on Vehicle exposure shows the description text, not the display key name.
CC-33163	The Damage Description in the property incident DV is now correct.
CC-33116	Printing a list view from Search Results no longer throws an exception.
CC-33040	Using EntityFactory to get a new Adjudicator entity now properly returns entity.Adjudicator.
CC-32957	Warnings from ContactCenter now show up in ClaimCenter.
CC-32760	The Claim Policy page (reached from the Policy action) now shows runtime exception messages correctly.
CC-32644	LossType in PolicySearchCriteria no longer returns null for verified policies when you enter the policy number manually without using the policy screen.
CC-32557	The Data Management System plugin is properly invoked to add a document back into the System; there is no null document error.
CC32291	Searching users by attribute in the Assignment Screen no longer causes an EvaluationException error.
CC-32181	User Administration screens now properly filter claims/exposures to show all of them, not just those assigned to the user.
CC-32132	The AddressBook Contact Detail page search returns contacts to ClaimCenter when it finds a contact with a long list of related contacts.
CC-32109	In the user interface, dropdown menus display correctly.
CC-32087	View and Edit buttons now remain when changing an edited draft document to 'final' status.
CC-32061	The Team and Admin tabs always return the same claim information for a user.
CC31989	Printing a large claim no longer causes the application server to fail.
CC-31775	Editing a draft claim always returns it to the New Claim wizard, even if you accessed one by typing its number in the Claim tab.
CC-31635	In the New Claim wizard, when three exposures are created and manual assignment of the claim and all three exposures is selected with three different adjusters, the exposures are now assigned to the right adjusters.
CC-31656	When two users change the same data object, the error received by the second user to save tells the name of the first user to have changed the object; this restores 3.1.x functionality.
CC-31473	You may now add multiple witnesses with the same name and case in the Witnesses LV on the Loss Details Page.
CC-30754	An update to Claim Loss no longer throws an error after adding a new claim association.
CC-30617	No exception is thrown when returning to assignment from the driving directions page.
CC-29364	Geocoding/Proximity search no longer fails if a 9-digit zip code is used.
CC-28689	Skipping 'pending' messages in the Admin tab no longer causes inconsistencies; related problems also fixed.

Issue ID	Description
CC-28403	Parties Involved page now refreshes properly after the error of changing a user role to claimant when another claimant already exists.
CC-19393	The Find Note Template screen now correctly refers to the DocumentType instead of the NoteType typelist.

Known Issues and Limitations

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

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Java JIT compiler options may cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's `-XX:CompileThreshold` option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance; however, the server may become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server may crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a plugin interface called `IApprovalAdapter`. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The `IApprovalAdapter` interface is not supported in ContactCenter.

Contact.Linked() and Contact.Synced() methods are missing (CC-27259)

Issue: These methods were intended to be deprecated, but were mistakenly removed completely.

Workaround: Change:

```
var linked = contact.Linked;

to

var linkStatusValue = contact.calculateLinkStatus().LinkStatusValue;
var linked = (linkStatusValue == "NotSynced" or
              linkStatusValue == "SyncedRemote" or
              linkStatusValue == "Synced");
```

and change:

```
var synced = contact.Synced;

to

var linkStatusValue = contact.calculateLinkStatus().LinkStatusValue;
var synced = (linkStatusValue == "Synced");
```

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or

- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Application doesn't respond when uploading a document with non-ASCII characters in its name (CC-28575)

Issue: When running on Tomcat, if you attempt to upload a document with non-ASCII characters in its name, the screen may show an hourglass cursor and not respond. This is a Tomcat configuration issue.

Workaround: In the `Tomcat/conf/server.xml` file, add the following attribute to the `<Connector>` definition in use: `useBodyEncodingForURI="true"`

Information about importing voided/stopped payments (CC-16750)

Issue: If you are importing voided/stopped payments on an exposure or claim that is closed, then you need to also be sure to generate necessary offsetting reserve transactions to keep open reserves at zero for the affected exposures/claims. The necessary offsetting payment for each voided/stopped payment will be auto-generated, but unless you add the offsetting reserve transactions to the staging table, you'll end up with open reserves on closed exposures/claims, and this will cause a consistency failure in the database. In addition, you cannot import a new exposure against an existing claim, nor can you import financials against an existing exposure/claim. These issues will be addressed in a future release.

You must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance. To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in `filename`.

Out of memory errors after reloading configuration many times (CC-28736)

Issue: After reloading the configuration files many times (perhaps 10-12 times), you may get out of memory errors.

Workaround: Restart the server.

Update to Claim Loss gives an error (CC-30754)

Issue: If you search for a claim without an association, then go to its **Loss Details** screen and add, then update, a claim association, you will receive the error "The Claim you are trying to update was changed by another user. Please cancel and retry your change."

Workaround: Make autosave locations false; this has the drawback that intermediate changes won't be saved when one moves between pages.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

Bulk Invoices created with Bulk Invoice API will not go from Awaiting Submission to Requesting when `bulkinvoicesescalation` is executed (CC-31567)

Issue: The `bulkinvoicesescalation` batch process now correctly escalates API-created bulk invoices on the correct date. However, if the time of the invoice is late, the batch process might miss the invoice until the next day.

Workaround: Set the `ScheduledSendDate` of the batch invoice to `null`.

IClaim depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: `IClaim` depends on `SynchStateData`, and `SynchStateData` uses deprecated functions like `setMessageSinkID` and `getMessageSinkID`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

Upgrading GScript resources overwrites all classes and libraries, but not rules (CC-31908)

Issue: There is a big difference between importing rules into Studio/resources vs. importing classes and libraries, namely that when you import classes and libraries, you overwrite what's already there. To prevent loss of your custom changes and additions to GScript classes and libraries additions, Guidewire no longer supplies libraries and classes export files with the release; the export files for rules are still present in `ClaimCenter4.0.3/` `Guidewire/ClaimCenter`.

Workaround: To safely import changed and new classes and libraries into this release, first compare the report provided in the “Notes on Upgrading Rules from Release 4.0.x” section of this readme document with your corresponding files. Then use Studio to transfer (copy and paste) the needed classes and libraries into your own resource files. Be careful to save the old version of your files before updating your GScript classes and libraries. The previous method of importing and merging is still useful for rules from older versions.

List of messages returned from `PendingMessages()` is not ordered correctly (CC-32661)

Issue: The list of messages returned from `MessageContext.PendingMessages()` isn't ordered by `SendOrder`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the **Checks** screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

Guidewire ClaimCenter 4.0.5 Release Notes

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

Release 4.0.5.146

Copyright © 2001-2008 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.5.146.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact technical support at support@guidewire.com.

Installing This Release

For general installation information, please refer to the *ClaimCenter Installation Guide*. The following issues pertain to installing this release in particular:

Changed Event Filtering Behavior For Messaging

This release contains a significant change in the way event messaging works.

WARNING This change may have a significant effect on your application behavior, one that may cause undesired behavior if it is not accounted for in your messaging configuration before you upgrade your production system. Before upgrading, you must make sure that you fully understand this change and the impact it may have on your application behavior. If you have any questions, contact your Guidewire customer support representative.

This issue requires your attention only if you have any “old-style” (that is, available since ClaimCenter 2.1) MessageSinks, and you also have the `DeprecatedEventFiltering` configuration parameter set to `false` (this is the default value, so unless you have explicitly set it to `true`, assume it is `false`). If you have no MessageSinks defined, or you have `DeprecatedEventFiltering` set to `true`, you will not be affected by this change.

The most likely scenario in which `DeprecatedEventFiltering` will be set to `false` is that you have a single ISO MessageSink defined, with other messaging tasks being handled by the new-style (available in ClaimCenter 4.0.0 and later) plugins. In this case, you should make sure to merge the ISO messaging rule changes in this 4.0.5 upgrade release; these handle both settings of `DeprecatedEventFiltering` correctly. If you have other MessageSinks defined besides ISO, and you have `DeprecatedEventFiltering` set to `false`, you will need to adjust your rules to handle `ClaimValid` and `ExposureValid` events correctly.

IMPORTANT If you are unsure how to handle your situation, contact your Guidewire customer support representative.

Explanation of the Change

In ClaimCenter 3.1.x and earlier, a destination was tied to a message sink, and when configuring a destination/message sink you could specify a validation level. The messaging layer would not invoke the messaging rules to process events for any destination until the entity on which the event occurred reached the required level. At that time, you would get an `xxxValid` event. So, for example, you might have a destination/message sink that was interested in claim changes, but only if the claim was valid at the `Ability to pay` level. In that case, the messaging layer would not invoke the messaging rules for that destination until the claim reached the `Ability to pay` validation level. When the claim did reach `Ability to pay`, the messaging rules will be called with a

ClaimValid event. After that they would be called with any subsequent ClaimChanged events. This was called *event filtering*, because the ClaimChanged events were filtered out until the claim reached the required validation level. Event filtering was implemented using a *synch state* array on claims and exposures, which recorded whether a given claim/exposure had sent a Valid event for a particular destination.

ClaimCenter 4.0.0 moved to the new messaging model, where destinations could be associated with transport plugins and specified in the messaging configuration (which was defined in `config.xml` at the time). It also dropped the idea of event filtering, assuming that it introduced a lot of complication and confusion but didn't add anything that couldn't already be handled in messaging rules. However, for backwards compatibility, ClaimCenter still supported destinations with message sinks. For such "old-style" message sink destinations (but not the new-style plugin-based destinations), ClaimCenter supported event filtering providing the special `DeprecatedEventFiltering` parameter was `true` (the default value is `false`).

To enforce this usage, ClaimCenter 4.0.4 and later removed support for this parameter, effectively forcing its value to always be `true`. (Setting the parameter to `false` was unlikely to result in correct behavior for an old-style message sink, unless its rules were explicitly modified to cope with the changed event filtering behavior.) A side effect of this change is that if you have the `DeprecatedEventFiltering` parameter set to its default value of `false`, then when upgrading to 4.0.5, your "old-style" message sinks would suddenly start receiving Valid events. Indeed, as no Valid events had been sent in the lifetime of the system, then pretty much any time a valid claim or exposure was touched it would generate a Valid event (because the *synch state* array was completely empty). This could lead to an "event flood" when a batch process touches a large number of items. Depending on the event messaging rules, this could in turn lead to a flood of messages to the corresponding destination.

Manually Upgrade the Rule Sets (CC-39604)

In 4.0.4, ClaimCenter introduced a new Transaction Pre-setup ruleset. If you are upgrading to 4.0.4 or 4.0.5 from earlier versions of either 3.1.x or 4.0.x, you will not have a `ruleset.xml` file. Studio and ClaimCenter need this file to execute this ruleset (even though it will start out empty, the file must at least exist). You must manually move a copy of this file to your upgraded configuration before running `upgrade.bat`:

To provide for the new Transaction Pre-setup rule set, you must manually add `ruleset.xml`:

1. Create the following new directory in your existing ClaimCenter installation: `ClaimCenter/config/resources/rules/PR/Transaction Pre-setup`.
2. In the same directory as these release notes, locate the file `ruleset.xml`.
3. Copy the `ruleset.xml` file into the new directory you created in step 1.

How to Use `itype` and `typeof` (CC-39157)

The `typeof` operator is the preferred way to get type information from an object. It operates with the run time type of the object. In contrast, the `itype` property on objects and other types represents the declared type (the type declared at compile time). This latter approach is discouraged and may be deprecated in a future release.

If you want information from a specific type known at compile time, simply access that type *directly* in GScript:

```
print(Integer.Name) // prints the String "java.lang.Integer"
```

In contrast, the `typeof` operator uses the run time type of the object:

```
print((typeof 29).Name) // prints the String "java.lang.Integer"
```

Staging Table Import (CC-35141)

All staging tables must contain a unique index on each table before loading the staging tables. In previous releases, no integrity constraints were performed during load/import automatically, only when integrity checks were run. However, now the system applies an integrity constraint during load that the staging tables is a unique index on all staging tables. If the table has a retired field, the unique index must be on the combined columns

(publicID, retired). If the table does not have a retired field, the unique index must be on the publicID column. This change was made in ClaimCenter 4.0.4 but was not mentioned in its release notes.

Install a New Reporting Module

A new build of InetSoft 8 is included in the 4.0.5 release. The new InetSoft build includes some issue fixes that have been requested.

If you are using the Reporting Module, you must install the new build of the InetSoft server with your upgrade to 4.0.5.

Best Practices for Upgrading Rules and Libraries

Since rules are now files kept in the `.../config/resources/rules` directory, the configuration upgrade process applies the same logic in evaluating your rules and the base and target configurations' rules to determine which rules to include in the upgraded configuration and which rules should be compared and merged. This is what the merge process tries to do:

- The merge process adds rules in the base configuration that doesn't exist in your configuration to your `...config/resources/rules` directory.
- Identical rules in both the base configuration and your configuration remain in this directory.
- When rules differ between the base configuration and your configuration, your rule remains in the `...config/resources/rules` directory, and the base rule is added to the merge directory.

Although this approach works well with other configuration files, it is not the optimal way for you to upgrade your rules, simply because you are likely to have modified most of the base rules with which you started. Thus, you will have many rules to merge with the target configuration.

The other major issue for merging rules is that rule execution order is important and is controlled by `ruleset.xml`. Unlike most other files, you should not edit or merge this file because the order of rule execution can drastically change the logic of rules. Studio should always generate this file.

Therefore, it is much easier and safer to start with your own, existing rules and then selectively apply to them the changes that Guidewire has made between the base and target configurations.

To Merge Rules Efficiently

1. Make a copy of your entire set of rules (in the `...config/resources/rules` directory) before running the automatic upgrader tool.
2. Copy your old `...config/resources/rules` directory to your upgraded configuration. Now the upgraded configuration contains all of your original rules.
3. Regenerate the toolkit and build a war file.
4. Point Studio to the upgraded configuration, which contains your old rules.
5. Review the rule differences between the base and target configurations to see a list of all the rule changes between the two releases.
6. Use Studio to import the updated `rules.xml` file.
7. Determine which rules to add (or modify and add) to your current rules based on the changes to the target.

Changes in this Release

This section describes the product changes in this release.

- Configuration Environment Changes

- Base PCF File Changes
- Base Resource Changes
- Improvements
- General Issues

Configuration Environment Changes

The following are the primary changes to the ClaimCenter configuration environment files, located in ClaimCenter/config:

File	Changes
display/base/display.properties	Added and modified several display keys.

Base PCF File Changes

To view a report of the changes in the base PCF files, [click here](#) (requires the readme_files directory on your local disk).

Base Resource Changes

The following links require the readme_files directory on your local disk.

- To view a report of the changes in the base resources from ClaimCenter release 4.0.0 to 4.0.5, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.1 to 4.0.5, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.2 to 4.0.5, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.3 to 4.0.5, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.4 to 4.0.5, [click here](#).

Notes on Upgrading Rules from Release 4.0.x

The upgrade to ClaimCenter 4.0.5 does not automatically merge in the changes that you have made to rules in previous 4.0.x releases. Therefore, we recommend that you preserve your customized rules and then, where desired, manually incorporate any changes that Guidewire made to the shipped rules provided with this release. To do so, use the following procedure:

1. Before upgrading, make a backup copy of your entire set of existing rules; that is, the complete ClaimCenter/config/resources/rules directory.
2. Run the upgrade tool.
3. Copy the rules directory from your backup to the upgraded ClaimCenter/config/resources directory. Allow the copy to overwrite any files that it needs.
4. Run Guidewire Studio. Import the rule changes from one (only one!) of the following files supplied with this ClaimCenter release:
 - If upgrading from 4.0.0, import the file: ClaimCenter Rule Diffs 400-405 For Import.xml
 - If upgrading from 4.0.1, import the file: ClaimCenter Rule Diffs 401-405 For Import.xml
 - If upgrading from 4.0.2, import the file: ClaimCenter Rule Diffs 402-405 For Import.xml
 - If upgrading from 4.0.3, import the file: ClaimCenter Rule Diffs 403-405 For Import.xml
 - If upgrading from 4.0.4, import the file: ClaimCenter Rule Diffs 403-405 For Import.xml

After you import the file, you will see all changed rules appear in Studio, disabled, and under parents named for the date of the import.

5. Review the rule differences detailed in the “Base Resource Changes” section of this Release Notes document.
6. If you wish to incorporate any of the rule changes into your own rules, do this in Studio. You may either copy the changed section from the imported rule into your rule, or you may delete your rule and replace it with the imported rule. When you are finished, delete any remaining unused imported rules.

WARNING You should make the changes only within Studio. Do not attempt to modify the rules XML files directly, or to merge the diffs using a third-party utility.

Improvements

The following are the primary improvements made to ClaimCenter for this release:

Issue ID	Description
CC-18489	After the security parameter <code>sessiontimeoutsecs</code> expires, no exception occurs when creating a note by selecting a note template.
CC-29233	There is no longer the restriction that Excluded Parties and Covered Parties have no Contacts in common.
CC-32995	The Pre-update rules are non-recursive, meaning that they do not run a second time on objects modified during execution of the Pre-update rules. ClaimCenter simply adds any objects modified by the pre-update rules to the list of objects needing validation as described by the validation graph.
CC-35992	You now have the ability to load-balance SOAP requests. In order to do this, SOAP is non-conversational and can access non-conversational proxies.
CC-36114	ClaimCenter now handles both integer and decimal input in the same way: For example, commas entered in both data types are preserved.
CC-37451	The thread time-out is now configurable; it used to have a hard-coded value of 30 seconds.
Database Improvements	
CC-18300	Application Servers now handle disconnections to the database more gracefully. When multiple servers use the same database, and one fails, the other servers compensate properly.

General Issues

The following are the primary issues corrected in this release:

Issue ID	Description
Financial Issues	
CC-33996	ClaimCenter now provides correct support for modifying the summary amounts of already existing single and recurring checks in rules.
CC-34418	Available reserves are now recalculated after altering the amount of a recurring check that is pending approval in the Change Recurring Schedule user interface screen.
CC-34538	Editing a non-eroding recurrence no longer causes an assertion failure.
CC-34559	<code>TestAuthorityLimits()</code> for <code>TransactionSet</code> with multiple reserves works properly; if two small reserves are created, the second is no longer allowed if the total exceeds the user's authority limit.
CC-37206	When transferring a check, the payee contact is no longer duplicated when the new claim has the same contact information.
CC-37339	A warning during creation of recurring checks does not affect the check numbers of the recurrences.
GScript Issues (see also Studio Issues)	
CC-39221	An <code>EntityFactory</code> in rules issue has been fixed. In previous releases, there were rare circumstances in which modified or new entities in rules could co-mingle with modified (or new) entities created by Java plugins and be committed together when using <code>EntityFactory</code> .
CC-33649	
CC-36693	<code>Exposure.hasTransaction()</code> now takes into account the current exposure when attempting to find a match.
CC-37132	<code>ClaimFinancialsAPIImpl.addClaimFinancialsWithValidation()</code> now transitions <code>lifecyclestates</code> correctly.

Issue ID	Description
Integration Issues	
CC-34581	The provided Geocode plugin now changes the geocode status for PO Box addresses after it fails.
CC-35026, CC-37671	The CheckOnInquiry step in the metro report workflow handles an error status by going into WaitBeforeInquiry if the report is returns an error status. This allows the workflow to continue.
CC-37677	Navigation time in the Administration tab is normal when a configured Report server is down.
CC-37742	Profiler tags have been added for profiling plugin calls.
CC-38711	Calling the SOAP API with a byref user object no longer creates extra UserSettings objects in the database.
CC-39173	A limitation concerning adding claims during financials calculations has been removed: In previous releases, it was important for data integrity never to add new claims to ClaimCenter via the web service APIs while financials calculations were running. This affected both the addFNOL() and migrateClaim() methods of IClaimAPI. This limitation has been removed, so you can use addFNOL() and migrateClaim() methods independent of whether financials are running.
CC-39714	Users no longer are logged out of InetSoft running in ClaimCenter; they no longer must repeatedly log in.
Server Issues, including Database, Upgrade and Batch Process Issues	
CC-34317	The java.lang.IllegalStateException 'Attempt to access bean of type official with a null bundle' no longer appears.
CC-34490	A java.lang.ArrayStoreException patch fixes the ReferenceMap implementation in the WebSphere external conversion layer.
CC-35363	A problem has been fixed that caused large memory consumption when adding a FNOL claim with a new note. This was due to the Claim Snapshot entity consuming large amounts of memory.
CC-35365	Closing the New Claim wizard no longer creates a blank note attached to the claim.
CC-35854	The ccst_loadererror table now shows the correct rownumber value during the required role claim contact loader validation.
CC-36537	A malicious user may no longer enter harmful script into a Textarea field. Instead of being executed, any script is merely displayed.
CC-36622	The default prefetch size was sometimes too large for some bean loads/refreshes. The chunk size is now set to the number of expected results; this reduces the size in many cases and increases performance.
CC-37346	Log messages with JDBC URL now mask out the password for the production database.
CC-37545	After you extend the database with a new field, and then create new claims, their claim snapshots (and those of previously created claims) now display correctly.
CC-38710	The PolicySearchSampleDataAdapter runs properly with JDK 1.5.
CC-38719	DataSource Connections are now correctly handled.
CC-38958	The gather_table_stats method now explicitly sets the cascade argument.
CC-39484	registry.csv now contains the checksums for all earlier version files; the upgrader uses these checksums.
CC-39607	The Database Parameters page no longer causes a problem in the SQL Server JNDI.
CC-39633	Deployment of the application to Websphere no longer requires prior setup of JNDI datasource.
CC-39803	An issue has been fixed in which occasionally, an extension entity for contacts in ContactCenter caused finger-printing in ClaimCenter to wrongly throw a Concurrent Data Exception error.
CC-39808	An Oracle function no longer returns a column name as COLUMN_VALUE, when an InetSoft query expected the column name to be GROUPID. This caused empty Dashboard reports.
Upgrade Issue	
CC-34274	Incident-related exposure types are no longer in the exposeureTypes typelist; this no longer causes a consistency check exception in upgrading the database to 4.0.5.
CC-36728	Metro reporting, added in 4.0, defines five activity patterns. These are now transferred with an upgrade trigger.
User Interface	
CC-32532	Assignment.assignByUserAttributes does not function as expected. This is a round-robin assignment, but when it considers user attributes, it does not assign in the same order as round-robin assignment without considering user attributes.
CC-33408	Validation errors are now called correctly when performing a policy refresh or policy select.
CC-33597	Creation of FNOL activities no longer checks the exposure validation level.

Issue ID	Description
CC-33649	EntityFactory In Rules problem is fixed: in previous releases, there were rare circumstances in which modified or new entities in rules could co-mingle with modified (or new) entities created by Java plugins and be committed together when using EntityFactory. This can no longer occur.
CC-33715	Validation of a claim and all its exposures lists validation error under the associated exposure.
CC-33739	IUserAPI.AddUser no longer forces a Credential to be provided; it honors the <code>credentialName</code> parameter as explained in the API doc, creating a Credential with user name specified by <code>credentialName</code> , and a default password of "invalidPassword".
CC-33740	The Group Header is now present from Aging PDF report when it prints.
CC-33841	Session Variables now correctly maintain their values, despite other actions that occur in the user interface.
CC-33936	The View Logs functionality is again present in ClaimCenter's Internal Tools. This was also fixed in 4.0.4.
CC-34142	Status codes in the dropdown created by the <code><PolicyStatCodeFilterCriteriaInput></code> tag now appear in sorted order.
CC-34920	A null pointer exception error no longer occurs when you create a check and the exposure then fails a validation rule at the Load/Save level. To avoid running validation after creating such a check, see Known Issues for this item.
CC-35803	The Event Messages screen now displays all incomplete messages, even when there are more than 2100 of them.
CC-36157	When the <code>available</code> attribute is set to <code>false</code> on the <code>CheckedValuesToolBarButton</code> , the button is grayed out and the user cannot click on it. No event is generated if the user clicks the grayed-out button.
CC-36947	You may now add extended fields to the note temple, because its velocity template now accepts the actual Note object, rather than a print wrapper of the object. Printing notes is unaffected.
CC-37966	Several Javascript alert message boxes are now configurable for other languages using <code>display.properties</code> .
CC-38071	When validating a claim, validation errors displayed on the Validation Results panel are now correctly labeled as being on the current panel.
CC-38201	The tests run by <code>DocumentBaseImpl()</code> on removal of documents now run in the correct order.
CC-38659	When completing the New Claim wizard, multiple validation warning on multiple entities (without any hard errors) display properly and do not hinder exiting the wizard.
CC-39215	You may now refresh a policy once you have edited it.
CC-39725	Members of a group no longer see other groups' claim data in Dashboard tab reports.
CC-39759	Dashboard drilldown reports are now correct, even when they include a large number of groups.

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin configuration in ClaimCenter/config/config.xml.

Java JIT compiler options may cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's -XX:CompileThreshold option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance; however, the server may become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server may crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a plugin interface called IApprovalAdapter. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The IApprovalAdapter interface is not supported in ContactCenter.

Application doesn't respond when uploading a document with non-ASCII characters in its name (CC-28575)

Issue: When running on Tomcat, if you attempt to upload a document with non-ASCII characters in its name, the screen may show an hourglass cursor and not respond. This is a Tomcat configuration issue.

Workaround: In the Tomcat/conf/server.xml file, add the following attribute to the <Connector> definition in use: `useBodyEncodingForURI="true"`

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the <workqueue> element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance. To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Out of memory errors after reloading configuration many times (CC-28736)

Issue: After reloading the configuration files many times (perhaps 10-12 times), you may get out of memory errors.

Workaround: Restart the server.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

Bulk Invoices created with Bulk Invoice API will not go from Awaiting Submission to Requesting when `bulkinvoicesescalation` is executed (CC-31567)

Issue: The `bulkinvoicesescalation` batch process now correctly escalates API-created bulk invoices on the correct date. However, if the time of the invoice is late, the batch process might miss the invoice until the next day.

Workaround: Set the `ScheduledSendDate` of the batch invoice to `null`.

`IClaim` depends on `SynchStateData`, which uses deprecated methods (CC-31611)

Issue: `IClaim` depends on `SynchStateData`, and `SynchStateData` uses deprecated functions like `setMessageSinkID` and `getMessageSinkID`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

List of messages returned from `PendingMessages()` is not ordered correctly (CC-32661)

Issue: The list of messages returned from `MessageContext.PendingMessages()` isn't ordered by `SendOrder`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the **Checks** screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

`Transaction.addNewLineItem()` does not return the newly created `TransactionLineItem` (CC-34192)

Issue: In the 4.0.4 release, `Transaction.addNewLineItem()` became scriptable so that it could be called from within the new Transaction Pre-setup ruleset. However, this method currently has a `void` return type, so if a rule needs to make changes to the newly added `lineitem` it would need to get the `lineitem` array from the transaction and iterate through them to find the `lineitem` to modify. Clearly this is un-necessarily difficult for rule writers, so we should modify the method to return the newly created `lineitem` so that the rule can get a handle to it directly.

Workaround: See the Issue description above to workaround in 4.0.5, or use 5.0.x, which has fixed this.

There is no scriptable domain method on `CheckSet` to simplify adding a new Check (CC-34193)

Issue: There is no scriptable domain method on `CheckSet` to simplify the adding of a new check from rules. In theory, a rule could just create a new `Check` directly and add it to the `CheckSet`, so long as it then used the `Check.addNewPayment()` method to add a payment to the check, but that is a potentially error prone process if the rule neglects to set some of the fields on `Check`.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns and rerun the PCF Converter.

A null pointer exception error no longer occurs when you create a check and the exposure then fails a validation rule at the Load/Save level. (CC-34920)

Issue: If your installation keeps some exposures at a low validation level, but still wants to write checks on the exposure, validation rules will give an error.

Workaround: You can adjust the Validation Graph to prevent the validation rules from causing errors due to changed fields. Add this to the end of `extensions.xml` (fieldName may need to be `ClaimID` or `Claim`):

```
<validationTriggerOverrides>
  <validationTriggerOverride entityName="Exposure" fieldName="ClaimID" triggersValidation="true"/>
</validationTriggerOverrides>
```

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running IE 6 (or earlier) and you go to the ClaimCenter Desktop Activities page and repeatedly hit the Activities link, your IE memory usage increases dramatically.

Workaround: If you cannot prevent users from doing this, upgrade to IE 7.

After logout, a request to return to the login page creates a new session in Websphere (CC-36394)

Issue: On logout in WebSphere, the current session is destroyed but a new one is created. In ClaimCenter, logging out takes you back to the login page. Whenever a browser interacts with an application server, the server creates a session if none existed.

Workaround: On logout, go outside of the Guidewire application and enforce static content caching: the browser will reuse static objects in its cache (like the login page) and therefore minimize the unnecessary interactions with the application server. Alternately, have a web tier cache the static content. This is basically adding another caching layer (on top of the browser) and would reduce this phenomenon further.

GScript can confuse package names with variable names, causing a verification error (CC-36846)

Issue: PCF verification fails, and gives an incorrect error, if you use the same name for a variable that you have used for a package name.

Workaround: Use fully qualified package names.

Stopping the application server confuses the assignment engine (CC-37425)

Issue: If you stop and restart the application server, the assignment engine loses its pointer to the last record it was on.

Workaround: Avoid restarting the application server while the assignment engine runs.

Claimant bit not set correctly during upgrade from 3.1.4 (CC-38469)

Issue: Claimant bit denormalization was added to 4.0.x to make searches by claimant last name faster. This bit is not changed during an upgrade to 4.0.x.

Workaround: None.

An ISO library issue can create a first party payload from a third party exposure (CC-39126)

Issue: An error in the ISO.gs library creates a first party payload for a third party exposure.

Workaround: To correct the problem in the ISO.gs library, change this code:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||  
    exposure.ExposureType == "PersonalPropertyDamage") {  
    payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)
```

To:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||  
    exposure.ExposureType == "PersonalPropertyDamage") {  
    if (exposure.Claim.InsuredDenorm == exposure.ClaimantDenorm) {  
        payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)  
    } else {  
        payload = isoPayloadGenerator.generateEditableInjuryPayload(exposure, null,  
                                                                    exposure.Incident.Description, null)  
    }  
}
```

This causes the library, for a property exposure, to use the property payload for first party and the injury payload for third party.

Guidewire ClaimCenter 4.0.6 Release Notes

Release 4.0.6.7

Copyright © 2008 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire ContactCenter, Guidewire Insurance Suite, and the Guidewire logo are trademarks 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 documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.6.7.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact Guidewire Customer Support. You may contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For general installation information, please refer to the *ClaimCenter Installation Guide*. The following issues pertain to installing this release in particular:

Changes to ISO Validation Rules

The ISO GScript library and ISO **Validation** rule for exposures were changed to use `DetailedBodyPart` (instead of `PrimaryBodyPart`) for the ISO `ClaimsInjuredInfo/BodyPartCd` field.

This is a small change, but it affects validation rules. Before this change, to be ISO valid a bodily injury exposure had to have the `PrimaryBodyPart` set. After this change, the `DetailedBodyPart` must be set.

WARNING This means that existing exposures that used to be considered ISO valid (because they had a non-null `PrimaryBodyPart` but no `DetailedBodyPart`) will now no longer be valid. This may cause problems when editing or changing them.

If you use the base rules and have such existing exposures, you should consider allowing some kind of backward compatibility; for example, downgrading the validation error to a warning (or not enforcing it at all) for older exposures.

Manually Upgrade the Rule Sets (CC-39604)

In 4.0.4, ClaimCenter introduced a new Transaction Pre-setup rule set. If you are upgrading to 4.0.6 from earlier versions of either 3.1.x or 4.0.x, you will not have a `ruleset.xml` file. Studio and ClaimCenter need this file to execute this rule set (even though it will start out empty, the file must at least exist). You must manually move a copy of this file to your upgraded configuration before running `upgrade.bat`:

To provide for the new Transaction Pre-setup rule set, you must manually add `ruleset.xml`:

1. Create the following new directory in your existing ClaimCenter installation: `ClaimCenter/config/resources/rules/PR/Transaction Pre-setup`.
2. In the same directory as these release notes, locate the file `ruleset.xml`.
3. Copy the `ruleset.xml` file into the new directory you created in step 1.

Best Practices for Upgrading Rules and Libraries

Since rules are now files kept in the `.../config/resources/rules` directory, the configuration upgrade process applies the same logic in evaluating your rules and the base and target configurations' rules to determine which rules to include in the upgraded configuration and which rules should be compared and merged. This is what the merge process tries to do:

- The merge process adds rules in the base configuration that doesn't exist in your configuration to your `...config/resources/rules` directory.

- Identical rules in both the base configuration and your configuration remain in this directory.
- When rules differ between the base configuration and your configuration, your rule remains in the `...config/resources/rules` directory, and the base rule is added to the merge directory.

Although this approach works well with other configuration files, it is not the optimal way for you to upgrade your rules, simply because you are likely to have modified most of the base rules with which you started. Thus, you will have many rules to merge with the target configuration.

The other major issue for merging rules is that rule execution order is important and is controlled by `ruleset.xml`. Unlike most other files, you should not edit or merge this file because the order of rule execution can drastically change the logic of rules. Studio should always generate this file.

Therefore, it is much easier and safer to start with your own, existing rules and then selectively apply to them the changes that Guidewire has made between the base and target configurations.

To Merge Rules Efficiently

1. Make a copy of your entire set of rules (in the `...config/resources/rules` directory) before running the automatic upgrader tool.
2. Copy your old `...config/resources/rules` directory to your upgraded configuration. Now the upgraded configuration contains all of your original rules.
3. Regenerate the toolkit and build a war file.
4. Point Studio to the upgraded configuration, which contains your old rules.
5. Review the rule differences between the base and target configurations to see a list of all the rule changes between the two releases.
6. Use Studio to import the updated `rules.xml` file.
7. Determine which rules to add (or modify and add) to your current rules based on the changes to the target.

Changes in this Release

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Improvements
- General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 4.0.5 and ClaimCenter 4.0.6, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- `gw.api.*` GScript API
- typelists
- display keys

Base PCF File Changes

To view a report of the changes in the base PCF files, [click here](#) (requires the `readme_files` directory on your local disk).

Base Resource Changes

The following links require the `readme_files` directory on your local disk.

- To view a report of the changes in the base resources from ClaimCenter release 4.0.0 to 4.0.6, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.1 to 4.0.6, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.2 to 4.0.6, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.3 to 4.0.6, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.4 to 4.0.6, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.5 to 4.0.6, [click here](#).

Notes on Upgrading Rules from Release 4.0.x

The upgrade to ClaimCenter 4.0.6 does not automatically merge in the changes that you have made to rules in previous 4.0.x releases. Therefore, we recommend that you preserve your customized rules and then, where desired, manually incorporate any changes that Guidewire made to the shipped rules provided with this release. To do so, use the following procedure:

1. Before upgrading, make a backup copy of your entire set of existing rules; that is, the complete `ClaimCenter/config/resources/rules` directory.
2. Run the upgrade tool.
3. Copy the `rules` directory from your backup to the upgraded `ClaimCenter/config/resources` directory. Allow the copy to overwrite any files that it needs.
4. Run Guidewire Studio. Import the rule changes from one (only one!) of the following files supplied with this ClaimCenter release:
 - If upgrading from 4.0.0, import the file: `ClaimCenter Rule Diffs 400-406 For Import.xml`
 - If upgrading from 4.0.1, import the file: `ClaimCenter Rule Diffs 401-406 For Import.xml`
 - If upgrading from 4.0.2, import the file: `ClaimCenter Rule Diffs 402-406 For Import.xml`
 - If upgrading from 4.0.3, import the file: `ClaimCenter Rule Diffs 403-406 For Import.xml`
 - If upgrading from 4.0.4, import the file: `ClaimCenter Rule Diffs 404-406 For Import.xml`
 - If upgrading from 4.0.5, import the file: `ClaimCenter Rule Diffs 405-406 For Import.xml`

After you import the file, you will see all changed rules appear in Studio, disabled, and under parents named for the date of the import.

5. Review the rule differences detailed in the “Base Resource Changes” section of this Release Notes document.
6. If you wish to incorporate any of the rule changes into your own rules, do this in Studio. You may either copy the changed section from the imported rule into your rule, or you may delete your rule and replace it with the imported rule. When you are finished, delete any remaining unused imported rules.

WARNING You should make the changes only within Studio. Do not attempt to modify the rules XML files directly, or to merge the diffs using a third-party utility.

Improvements

The following are the primary improvements made to ClaimCenter for this release:

Issue ID	Description
CC-40470	Removed the phone number from the Driver1 element in a Metro request; this is not part of Metro's schema.
CC-40835	In a clustered environment, script parameters are no longer reloaded into the database when a non-batch server restarts.
CC-41753	Updated ISO stylesheet to latest version. Among other changes, this version masks out the first five digits of Social Security Numbers.
CC-41899	When initiating a round robin assignment, the first assignee is selected at random among the list of available assignees. This prevents those at the front of the list from receiving disproportionate assignments after a server restart or in a clustered environment.
CC-42153	You can now change the "Eroding" or "Non-Eroding" setting of a payment if its status is Submitted or Submitting.
CC-42724	The GuidewireDocumentAssistant ActiveX control is now signed, for greater security.
CC-43116	ContactCenter only: The default value for configuration parameter AllowActiveX has been changed to false.
CC-43425	The documentation and recommendations have changed regarding importing records from database tables (also known as staging table import). Once all records are loaded, you must run financials calculations again to properly update the report tables.

General Issues

The following are the primary issues corrected in this release:

Issue ID	Description
CC-38235	Fixed an issue where NewClaimWizardInfo (initialized in the initialValue of the "Wizard" variable) had a null pointer to the wizard.
CC-39037	Fixed an issue where a new activity was being created each time a response to a Metro report inquiry was deferred.
CC-39415	Fixed an error that resulted in a "NULL" message when you created certain combinations of authority limits.
CC-39574	Fixed an issue where a validation warning when creating a check would display as an error.
CC-39653	Update ISO validation rules to use DetailedBodyPart (instead of PrimaryBodyPart) for the ISO ClaimsInjuredInfo/BodyPartCd field. NOTE: This is a change to validation rules; ensure that this change does not cause currently valid data to become invalid. For more information, see "Changes to ISO Validation Rules" on page 262.
CC-40070	Fixed an error in the Note.vm template used for printing notes.
CC-40079	Fixed an issue where setting the InputHelpTextOnFocus configuration parameter to false would display "undefined" when a field was in focus.
CC-40250	The Metro FireAutoReport now has a request type of X.
CC-40651	Fixed an issue where validation warnings would be displayed in the results page after they were fixed.
CC-40842	Fixed an issue where the Activity Patterns list on the Admin tab would remain in English in a localised application.
CC-40949	Fixed an error when more than one exposure was created within a rule.
CC-41047	Fixed an issue where ClaimCenter would send a TaxID to ISO in the wrong format.
CC-41268	Fixed an issue where it would take a long time to load contacts on a claim.
CC-41374	Fixed an issue where deleting an authority limit profile would not delete the corresponding authority limits.
CC-41446	Fixed an issue where an Imported check that qualifies as a duplicate could have a repeated "This check is a potential duplicate check." message in its approval result.
CC-41602	Fixed an issue where a data load would get stuck on insertion into cc_transaction.
CC-41615	ISO integration now includes contacts owned by an exposure's incident, in addition to those owned directly by the exposure.

Issue ID	Description
CC-41961	Fixed an issue where the Team screen showed a negative number for Other.
CC-42050	Corrected the following behavior with financials-related methods: <ul style="list-style-type: none"> • <code>setPaymentType()</code> now changes <code>NonEroding</code> to false if the payment type is not Supplemental. • <code>setNonEroding()</code> now attempts to set its associated check's payment to eroding or non-eroding as long as the check exists.
CC-42429	Fixed an issue on the Claim Search screen where the Search For Date and Financial Value fields would show only the value that was previously chosen and <none selected> as options.
CC-42444	Fixed an error that would occur when creating a check with an empty Service Period Start value.
CC-42625	Fixed an issue where authority limits would not be checked properly for small amounts.
CC-42658	Fixed an issue where a text box containing a newline would display <> characters.
CC-42679	Fixed an issue where a claim search with the loss date of "today" as a date range would not return that days' claims.
CC-42727	Fixed an issue where a document would not be shown when a "No Report" response was received from Metro.
CC-42931	Fixed an error that would occur if you try to create a new reserve, get a validation error, do not clear the validation error, and click Add to try to add another reserve.
CC-42428	Fixed an exception that was thrown when a coverage type was changed on an existing aggregate limit to any coverage type that had possible covered items but none were selected.
CC-43299	Fixed an exception that would occur when creating an exposure.
CC-43366	When transferring a check or recoding a payment, the <code>PaymentType</code> and <code>DoesNotErodeReserve</code> (Eroding vs. Non-eroding) properties were overwritten during commit to be the same as the original payments. This hard coding has been removed. You can now edit these values (if added to the appropriate PCFs) during the recode or transfer operation (this will still force the Supplement payment type if the target Exposure is closed). It is also possible to add GScript to the action attribute of the Transfer or Recode button to modify the properties before the new transactions are saved. See also CC-43547 in "Known Issues and Limitations" on page 266.

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess]
```

registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Java JIT compiler options may cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's `-XX:CompileThreshold` option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance; however, the server may become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server may crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a plugin interface called `IApprovalAdapter`. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The `IApprovalAdapter` interface is not supported in ContactCenter.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance. To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Out of memory errors after reloading configuration many times (CC-28736)

Issue: After reloading the configuration files many times (perhaps 10-12 times), you may get out of memory errors.

Workaround: Restart the server.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaim depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: `IClaim` depends on `SynchStateData`, and `SynchStateData` uses deprecated functions like `setMessageSinkID` and `getMessageSinkID`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

List of messages returned from PendingMessages() is not ordered correctly (CC-32661)

Issue: The list of messages returned from `MessageContext.PendingMessages()` isn't ordered by `SendOrder`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the **Checks** screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

Transaction.addNewLineItem() does not return the newly created TransactionLineItem (CC-34192)

Issue: In the 4.0.4 release, Transaction.addNewLineItem() became scriptable so that it could be called from within the new Transaction Pre-setup ruleset. However, this method currently has a void return type, so if a rule needs to make changes to the newly added lineitem it would need to get the lineitem array from the transaction and iterate through them to find the lineitem to modify. Clearly this is un-necessarily difficult for rule writers, so we should modify the method to return the newly created lineitem so that the rule can get a handle to it directly.

Workaround: See the Issue description above to workaround in 4.0.5, or use 5.0.x, which has fixed this.

There is no scriptable domain method on CheckSet to simplify adding a new Check (CC-34193)

Issue: There is no scriptable domain method on CheckSet to simplify the adding of a new check from rules. In theory, a rule could just create a new Check directly and add it to the CheckSet, so long as it then used the Check.addNewPayment() method to add a payment to the check, but that is a potentially error prone process if the rule neglects to set some of the fields on Check.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns and rerun the PCF Converter.

A null pointer exception error no longer occurs when you create a check and the exposure then fails a validation rule at the Load/Save level. (CC-34920)

Issue: If your installation keeps some exposures at a low validation level, but still wants to write checks on the exposure, validation rules will give an error.

Workaround: You can adjust the Validation Graph to prevent the validation rules from causing errors due to changed fields. Add this to the end of extensions.xml (fieldName may need to be ClaimID or Claim):

```
<validationTriggerOverrides>
  <validationTriggerOverride entityName="Exposure" fieldName="ClaimID" triggersValidation="true"/>
</validationTriggerOverrides>
```

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running IE 6 (or earlier) and you go to the ClaimCenter Desktop Activities page and repeatedly hit the Activities link, your IE memory usage increases dramatically.

Workaround: If you cannot prevent users from doing this, upgrade to IE 7.

GScript can confuse package names with variable names, causing a verification error (CC-36846)

Issue: PCF verification fails, and gives an incorrect error, if you use the same name for a variable that you have used for a package name.

Workaround: Use fully qualified package names.

Claimant bit not set correctly during upgrade from 3.1.4 (CC-38469)

Issue: Claimant bit denormalization was added to 4.0.x to make searches by claimant last name faster. This bit is not changed during an upgrade to 4.0.x.

Workaround: None.

An ISO library issue can create a first party payload from a third party exposure (CC-39126)

Issue: An error in the ISO.gs library creates a first party payload for a third party exposure.

Workaround: To correct the problem in the ISO.gs library, change this code:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||
    exposure.ExposureType == "PersonalPropertyDamage") {
    payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)
```

To:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||
    exposure.ExposureType == "PersonalPropertyDamage") {
    if (exposure.Claim.InsuredDenorm == exposure.ClaimantDenorm) {
        payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)
    } else {
        payload = isoPayloadGenerator.generateEditableInjuryPayload(exposure, null,
            exposure.Incident.Description, null)
    }
}
```

This causes the library, for a property exposure, to use the property payload for first party and the injury payload for third party.

Unchanged typelist files are marked as requiring a merge during upgrade (CC-43457)

Issue: The upgrader normally marks changed typelist files as needing to be merged. However, the upgrader may incorrectly mark end-of-line characters as having changed.

Workaround: Review the suggested merges, and ignore any that are based solely on the end-of-line character changing.

Removed hardcoding of PaymentType and NonEroding during recode and transfer (CC-43547)

Issue: When transferring or recoding a Supplemental payment to an open exposure, the PaymentType must change to Partial or Final. Hard coded logic would set the new payment to have a PaymentType of Partial, but it set the new payment's NonEroding flag (DoesNotErodeReserves) to have the same value as the original payment, which for a Supplement is NonEroding. Most Partial and Final payments are Eroding, so this behavior is counterintuitive and may not have been anticipated during implementation. However, it could have been anticipated and relied upon for a solution, so the behavior has been left unchanged for backwards compatibility.

Workaround: Now that this has been fixed (see CC-43366) and the hard coding removed, you can fix this behavior through GScript and PCF changes. Please contact Guidewire Support for assistance.

Guidewire ClaimCenter 4.0.7 Release Notes

Release 4.0.7.7

Copyright © 2008 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire ContactCenter, Guidewire Insurance Suite, and the Guidewire logo are trademarks 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 documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.7.7.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact Guidewire Customer Support. You may contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For general installation information, please refer to the *ClaimCenter Installation Guide*. The following issues pertain to installing this release in particular:

Manually Upgrade the Rule Sets (CC-39604)

In 4.0.4, ClaimCenter introduced a new Transaction Pre-setup rule set. If you are upgrading to 4.0.7 from earlier versions of either 3.1.x or 4.0.x, you will not have a `ruleset.xml` file. Studio and ClaimCenter need this file to execute this rule set (even though it will start out empty, the file must at least exist). You must manually move a copy of this file to your upgraded configuration before running `upgrade.bat`:

To provide for the new Transaction Pre-setup rule set, you must manually add `ruleset.xml`:

1. Create the following new directory in your existing ClaimCenter installation: `ClaimCenter/config/resources/rules/PR/Transaction Pre-setup`.
2. In the same directory as these release notes, locate the file `ruleset.xml`.
3. Copy the `ruleset.xml` file into the new directory you created in step 1.

Best Practices for Upgrading Rules and Libraries

Since rules are now files kept in the `.../config/resources/rules` directory, the configuration upgrade process applies the same logic in evaluating your rules and the base and target configurations' rules to determine which rules to include in the upgraded configuration and which rules should be compared and merged. This is what the merge process tries to do:

- The merge process adds rules in the base configuration that doesn't exist in your configuration to your `...config/resources/rules` directory.
- Identical rules in both the base configuration and your configuration remain in this directory.
- When rules differ between the base configuration and your configuration, your rule remains in the `...config/resources/rules` directory, and the base rule is added to the merge directory.

Although this approach works well with other configuration files, it is not the optimal way for you to upgrade your rules, simply because you are likely to have modified most of the base rules with which you started. Thus, you will have many rules to merge with the target configuration.

The other major issue for merging rules is that rule execution order is important and is controlled by `ruleset.xml`. Unlike most other files, you should not edit or merge this file because the order of rule execution can drastically change the logic of rules. Studio should always generate this file.

Therefore, it is much easier and safer to start with your own, existing rules and then selectively apply to them the changes that Guidewire has made between the base and target configurations.

To Merge Rules Efficiently

1. Make a copy of your entire set of rules (in the `...config/resources/rules` directory) before running the automatic upgrader tool.
2. Copy your old `...config/resources/rules` directory to your upgraded configuration. Now the upgraded configuration contains all of your original rules.
3. Regenerate the toolkit and build a war file.
4. Point Studio to the upgraded configuration, which contains your old rules.
5. Review the rule differences between the base and target configurations to see a list of all the rule changes between the two releases.
6. Use Studio to import the updated rules.xml file.
7. Determine which rules to add (or modify and add) to your current rules based on the changes to the target.

Changes in this Release

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Improvements and General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 4.0.6 and ClaimCenter 4.0.7, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- `gw.api.*` GScript API
- typelists
- display keys

Base PCF File Changes

To view a report of the changes in the base PCF files, [click here](#) (requires the `readme_files` directory on your local disk).

Base Resource Changes

The following links require the `readme_files` directory on your local disk.

- To view a report of the changes in the base resources from ClaimCenter release 4.0.0 to 4.0.7, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.1 to 4.0.7, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.2 to 4.0.7, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.3 to 4.0.7, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.4 to 4.0.7, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.5 to 4.0.7, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.6 to 4.0.7, [click here](#).

Notes on Upgrading Rules from Release 4.0.x

The upgrade to ClaimCenter 4.0.7 does not automatically merge in the changes that you have made to rules in previous 4.0.x releases. Therefore, we recommend that you preserve your customized rules and then, where desired, manually incorporate any changes that Guidewire made to the shipped rules provided with this release. To do so, use the following procedure:

1. Before upgrading, make a backup copy of your entire set of existing rules; that is, the complete ClaimCenter/config/resources/rules directory.
2. Run the upgrade tool.
3. Copy the rules directory from your backup to the upgraded ClaimCenter/config/resources directory. Allow the copy to overwrite any files that it needs.
4. Run Guidewire Studio. Import the rule changes from one (only one!) of the following files supplied with this ClaimCenter release:
 - If upgrading from 4.0.0, import the file: ClaimCenter Rule Diffs 400-407 For Import.xml
 - If upgrading from 4.0.1, import the file: ClaimCenter Rule Diffs 401-407 For Import.xml
 - If upgrading from 4.0.2, import the file: ClaimCenter Rule Diffs 402-407 For Import.xml
 - If upgrading from 4.0.3, import the file: ClaimCenter Rule Diffs 403-407 For Import.xml
 - If upgrading from 4.0.4, import the file: ClaimCenter Rule Diffs 404-407 For Import.xml
 - If upgrading from 4.0.5, import the file: ClaimCenter Rule Diffs 405-407 For Import.xml
 - If upgrading from 4.0.5, import the file: ClaimCenter Rule Diffs 406-407 For Import.xml

After you import the file, you will see all changed rules appear in Studio, disabled, and under parents named for the date of the import.

5. Review the rule differences detailed in the “Base Resource Changes” section of this Release Notes document.
6. If you wish to incorporate any of the rule changes into your own rules, do this in Studio. You may either copy the changed section from the imported rule into your rule, or you may delete your rule and replace it with the imported rule. When you are finished, delete any remaining unused imported rules.

WARNING You should make the changes only within Studio. Do not attempt to modify the rules XML files directly, or to merge the diffs using a third-party utility.

Improvements and General Issues

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

ID	Description
Assignment	
CC-46785	Fixed an error that occurred when calling assignUserByProximityWithSearchCriteria().
ContactCenter Integration	
CC-47157	Fixed an issue where ClaimCenter could not display the details of a contact created in ContactCenter.
Database Upgrader	
CC-45604	Fixed an issue where upgrading from ClaimCenter 3.1 to ClaimCenter 4.0.3-4.0.6 would cause the ClaimContact.ClaimantFlag property to be set incorrectly. When upgrading to ClaimCenter 4.0.7, this error is corrected.
Document Management	
CC-45859	To improve document management performance, ClaimCenter has changed the file structure for document storage. New documents created after upgrading will use this new structure; however the location of existing documents will not change. If you are having performance issues with the old document structure and you would like to migrate your existing documents to the new structure, please contact Guidewire Customer Support for assistance.

ID	Description
Documentation	
CC-45710	When using Sun JVM with WebLogic, add the -server flag as an argument when launching the WebLogic start script. See the Installation Guide for details.
Financials	
CC-43423	Fixed an issue in the New Check wizard where changing the payment type from Supplement to Partial would not change the Non-Eroding property back to Eroding.
CC-44175	Improved the performance of a query for retrieving claim financial transactions in the file ClaimFinancial-sTransactions.pcf.
Metropolitan Police Reports	
CC-44376	When sending an auto accident report to Metro, the request now contains the vehicle information (make, model, and so on) if it is available.
Reporting subsystem	
CC-43878	Removed blank lines in the Oracle scripts for reporting. These would sometimes cause errors.

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of #####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Java JIT compiler options may cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's `-XX:CompileThreshold` option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance; however, the server may become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server may crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a plugin interface called `IApprovalAdapter`. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The `IApprovalAdapter` interface is not supported in ContactCenter.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance. To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Out of memory errors after reloading configuration many times (CC-28736)

Issue: After reloading the configuration files many times (perhaps 10-12 times), you may get out of memory errors.

Workaround: Restart the server.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaim depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: `IClaim` depends on `SynchStateData`, and `SynchStateData` uses deprecated functions like `setMessageSinkID` and `getMessageSinkID`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

List of messages returned from PendingMessages() is not ordered correctly (CC-32661)

Issue: The list of messages returned from `MessageContext.PendingMessages()` isn't ordered by `SendOrder`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the **Checks** screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

Transaction.addNewLineItem() does not return the newly created TransactionLineItem (CC-34192)

Issue: In the 4.0.4 release, `Transaction.addNewLineItem()` became scriptable so that it could be called from within the new Transaction Pre-setup ruleset. However, this method currently has a `void` return type, so if a rule needs to make changes to the newly added `lineitem` it would need to get the `lineitem` array from the transaction and iterate through them to find the `lineitem` to modify. Clearly this is un-necessarily difficult for rule writers, so we should modify the method to return the newly created `lineitem` so that the rule can get a handle to it directly.

Workaround: See the Issue description above to workaround in 4.0.5, or use 5.0.x, which has fixed this.

There is no scriptable domain method on CheckSet to simplify adding a new Check (CC-34193)

Issue: There is no scriptable domain method on CheckSet to simplify the adding of a new check from rules. In theory, a rule could just create a new Check directly and add it to the CheckSet, so long as it then used the Check.addNewPayment() method to add a payment to the check, but that is a potentially error prone process if the rule neglects to set some of the fields on Check.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns and rerun the PCF Converter.

A null pointer exception error no longer occurs when you create a check and the exposure then fails a validation rule at the Load/Save level. (CC-34920)

Issue: If your installation keeps some exposures at a low validation level, but still wants to write checks on the exposure, validation rules will give an error.

Workaround: You can adjust the Validation Graph to prevent the validation rules from causing errors due to changed fields. Add this to the end of extensions.xml (fieldName may need to be ClaimID or Claim):

```
<validationTriggerOverrides>
  <validationTriggerOverride entityName="Exposure" fieldName="ClaimID" triggersValidation="true"/>
</validationTriggerOverrides>
```

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running IE 6 (or earlier) and you go to the ClaimCenter Desktop Activities page and repeatedly hit the Activities link, your IE memory usage increases dramatically.

Workaround: If you cannot prevent users from doing this, upgrade to IE 7.

GScript can confuse package names with variable names, causing a verification error (CC-36846)

Issue: PCF verification fails, and gives an incorrect error, if you use the same name for a variable that you have used for a package name.

Workaround: Use fully qualified package names.

Claimant bit not set correctly during upgrade from 3.1.4 (CC-38469)

Issue: Claimant bit denormalization was added to 4.0.x to make searches by claimant last name faster. This bit is not changed during an upgrade to 4.0.x.

Workaround: None.

An ISO library issue can create a first party payload from a third party exposure (CC-39126)

Issue: An error in the ISO.gs library creates a first party payload for a third party exposure.

Workaround: To correct the problem in the ISO.gs library, change this code:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||
    exposure.ExposureType == "PersonalPropertyDamage") {
    payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)
```

To:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||
```

```
        exposure.ExposureType == "PersonalPropertyDamage") {  
    if (exposure.Claim.InsuredDenorm == exposure.ClaimantDenorm) {  
        payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)  
    } else {  
        payload = isoPayloadGenerator.generateEditableInjuryPayload(exposure, null,  
            exposure.Incident.Description, null)  
    }  
}
```

This causes the library, for a property exposure, to use the property payload for first party and the injury payload for third party.

Unchanged typelist files are marked as requiring a merge during upgrade (CC-43457)

Issue: The upgrader normally marks changed typelist files as needing to be merged. However, the upgrader may incorrectly mark end-of-line characters as having changed.

Workaround: Review the suggested merges, and ignore any that are based solely on the end-of-line character changing.

Removed hardcoding of PaymentType and NonEroding during recode and transfer (CC-43547)

Issue: When transferring or recoding a Supplemental payment to an open exposure, the PaymentType must change to Partial or Final. Hard coded logic would set the new payment to have a PaymentType of Partial, but it set the new payment's NonEroding flag (DoesNotErodeReserves) to have the same value as the original payment, which for a Supplement is NonEroding. Most Partial and Final payments are Eroding, so this behavior is counterintuitive and may not have been anticipated during implementation. However, it could have been anticipated and relied upon for a solution, so the behavior has been left unchanged for backwards compatibility.

Workaround: Now that this has been fixed and the hard coding removed, you can fix this behavior through GScript and PCF changes. Please contact Guidewire Support for assistance.

Guidewire ClaimCenter 4.0.8 Release Notes

Release 4.0.8.9

Copyright © 2009 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire ContactCenter, Guidewire Insurance Suite, and the Guidewire logo are trademarks 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 documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 4.0.8.9.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. You can contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For general installation information, refer to the *ClaimCenter Installation Guide*. The following issues pertain to installing this release in particular:

Manually Upgrade the Rule Sets (CC-39604)

In 4.0.4, ClaimCenter introduced a new Transaction Pre-setup rule set. If you are upgrading to 4.0.8 from earlier versions of either 3.1.x or 4.0.x, you will not have a `ruleset.xml` file. Studio and ClaimCenter need this file to execute this rule set (even though it will start out empty, the file must at least exist). You must manually move a copy of this file to your upgraded configuration before running `upgrade.bat`:

To provide for the new Transaction Pre-setup rule set, you must manually add `ruleset.xml`:

1. Create the following new directory in your existing ClaimCenter installation: `ClaimCenter/config/resources/rules/PR/Transaction Pre-setup`.
2. In the same directory as these release notes, locate the file `ruleset.xml`.
3. Copy the `ruleset.xml` file into the new directory you created in step 1.

Best Practices for Upgrading Rules and Libraries

Since rules are now files kept in the `.../config/resources/rules` directory, the configuration upgrade process applies the same logic in evaluating your rules and the base and target configurations' rules to determine which rules to include in the upgraded configuration and which rules should be compared and merged. This is what the merge process tries to do:

- The merge process adds rules in the base configuration that do not exist in your configuration to your `...config/resources/rules` directory.
- Identical rules in both the base configuration and your configuration remain in this directory.
- When rules differ between the base configuration and your configuration, your rule remains in the `...config/resources/rules` directory, and the base rule is added to the merge directory.

Although this approach works well with other configuration files, it is not the optimal way for you to upgrade your rules, simply because you are likely to have modified most of the base rules with which you started. Thus, you will have many rules to merge with the target configuration.

The other major issue for merging rules is that rule execution order is important and is controlled by `ruleset.xml`. Unlike most other files, you should not edit or merge this file because the order of rule execution can drastically change the logic of rules. Studio should always generate this file.

Therefore, it is easier and safer to start with your own, existing rules and then selectively apply to them the changes that Guidewire has made between the base and target configurations.

To Merge Rules Efficiently

1. Make a copy of your entire set of rules (in the `...config/resources/rules` directory) before running the automatic upgrader tool.
2. Copy your old `...config/resources/rules` directory to your upgraded configuration. Now the upgraded configuration contains all of your original rules.
3. Regenerate the toolkit and build a `.war` file.
4. Point Studio to the upgraded configuration, which contains your old rules.
5. Review the rule differences between the base and target configurations to see a list of all the rule changes between the two releases.
6. Use Studio to import the updated `rules.xml` file.
7. Determine which rules to add (or modify and add) to your current rules based on the changes to the target.

Changes in this Release

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Improvements and General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 4.0.7 and ClaimCenter 4.0.8, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- `gw.api.*` GScript API
- typelists
- display keys

Base PCF File Changes

To view a report of the changes in the base PCF files, [click here](#) (requires the `readme_files` directory on your local disk).

Base Resource Changes

The following links require the `readme_files` directory on your local disk.

- To view a report of the changes in the base resources from ClaimCenter release 4.0.0 to 4.0.8, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.1 to 4.0.8, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.2 to 4.0.8, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.3 to 4.0.8, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.4 to 4.0.8, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.5 to 4.0.8, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.6 to 4.0.8, [click here](#).

- To view a report of the changes in the base resources from ClaimCenter release 4.0.7 to 4.0.8, [click here](#).

Notes on Upgrading Rules from Release 4.0.x

The upgrade to ClaimCenter 4.0.8 does not automatically merge in the changes that you have made to rules in previous 4.0.x releases. Therefore, we recommend that you preserve your customized rules and then, where desired, manually incorporate any changes that Guidewire made to the shipped rules provided with this release. To do so, use the following procedure:

1. Before upgrading, make a backup copy of your entire set of existing rules; that is, the complete ClaimCenter/config/resources/rules directory.
2. Run the upgrade tool.
3. Copy the rules directory from your backup to the upgraded ClaimCenter/config/resources directory. Allow the copy to overwrite any files that it needs.
4. Run Guidewire Studio. Import the rule changes from one (only one!) of the following files supplied with this ClaimCenter release:
 - If upgrading from 4.0.0, import the file: ClaimCenter Rule Diffs 400-408 For Import.xml
 - If upgrading from 4.0.1, import the file: ClaimCenter Rule Diffs 401-408 For Import.xml
 - If upgrading from 4.0.2, import the file: ClaimCenter Rule Diffs 402-408 For Import.xml
 - If upgrading from 4.0.3, import the file: ClaimCenter Rule Diffs 403-408 For Import.xml
 - If upgrading from 4.0.4, import the file: ClaimCenter Rule Diffs 404-408 For Import.xml
 - If upgrading from 4.0.5, import the file: ClaimCenter Rule Diffs 405-408 For Import.xml
 - If upgrading from 4.0.6, import the file: ClaimCenter Rule Diffs 406-408 For Import.xml
 - If upgrading from 4.0.7, import the file: ClaimCenter Rule Diffs 407-408 For Import.xml

After you import the file, you will see all changed rules appear in Studio, disabled, and under parents named for the date of the import.

5. Review the rule differences detailed in the “Base Resource Changes” section of this Release Notes document.
6. If you want to incorporate any of the rule changes into your own rules, do this in Studio. You can either copy the changed section from the imported rule into your rule, or delete your rule and replace it with the imported rule. When you are finished, delete any remaining unused imported rules.

WARNING You should make the changes only within Studio. Do not attempt to modify the rules XML files directly, or to merge the diffs using a third-party utility.

Improvements and General Issues

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

ID	Description
Financials	
CC-46897	Corrected an issue where you were unable to void a transferred check, when the system was configured to disallow multiple payments on a check (AllowMultiplePayments=false).
CC-47221	Corrected an issue where a payment with PaymentType = Supplement must have DoesNotErodeReserves = true. Saving a Supplement payment that is eroding is not allowed, whether created through rules or the IClaimFinancials API.

ID	Description
CC-48049	Fixed an issue where the method <code>IClaimFinancialsAPI.updateCheckStatus</code> now verifies that the current check status is at least a committed status, before applying the new status. This prevents mistaken updates to <i>PendingApproval</i> or <i>AwaitingSubmission</i> checks. However, this could be even more restrictive, so some invalid status updates will still not be prevented. Refer to the ClaimCenter Application Guide for valid check status transitions.
CC-48075	Fixed an issue where the sum on the Set Reserves page was not correctly calculated on the Change column.
CC-48124	Fixed an issue where the <code>FinancialCalculations</code> batch process calculated the gross amount in the <code>cc_checkrpt</code> table incorrectly for checks that are recoded and then stopped.
CC-48745	Corrected an issue where you were unable to void a transferred check, when the system was configured to disallow multiple payments on a check (<code>AllowMultiplePayments=false</code>).
General	
CC-47424	Fixed a problem adding up the summary row total for a numerical list view column, which could result in the display of too many fractional digits.
CC-47610	Fixed an issue where multiple spaces in a <i>textarea</i> are rendered as <code>&nbsp;</code> . The browser sent these back as ascii 160's, which were committed to the database.
CC-47804	Updated the Javadoc about the running of validation for the method: <code>IClaimFinancialsAPI.addClaimFinancials</code> .
CC-47896	Fixed an issue regarding a performance regression when you accessed the Event Message screen through the Administration tab. The fix for this included adding two new indexes to the message table. Some customers who are upgrading from CC4.0.x to CC4.0.8 have already been asked to add these indexes through the <code>extensions.xml</code> file. Before upgrading to 4.0.8, they should remove the indexes from the <code>extensions.xml</code> file and bump up the extension version.
History	
CC-47302	Fixed an issue where a litigation history event is no longer created when the Claim's litigation status is set to <i>Not Litigated</i> .
ISO Integration	
CC-45760	Fixed an issue so that you can request that no search is done when sending updates to ISO.
Logging	
CC-30190	Fixed an issue where a server memory amount greater than 2GB was printed incorrectly in the log file.
Metadata Issues	
CC-48497	Fixed a bug where defining an array on Incident of the Exposures pointing to the Incident caused a stack-trace at runtime while trying to create an exposure on any type of claim. The bug only affected this particular Exposure-Incident relationship.
Reporting Subsystem	
CC-47130	Fixed an issue where if you clicked Overdue Activities in the user interface with a specific claim number, the system returned results for the claim but ignored the claim number.
CC-47131	Corrected an issue in the reporting module where entering an invalid policy number in the Loss Run returned a blank screen. The fix included adding a message when no records were returned.
CC-47354	Corrected an issue in the reporting module where the <i>Current Catastrophe Financials Drill Down Days from Closed to Reopened</i> data was incorrect. The fix included putting in logic to only show reopen values for Claims with a status of <i>open</i> .
CC-47463	Fixed an issue in the reporting module where the <i>First payment productivity</i> report returned the error: Group columns defined in group filter not found . This error was due to alias' not being in place for the columns in the query being used for the <i>First Payment</i> report. The alias' have been added to the query for Oracle.
Server	
CC-49005	Removed the unused <code>ant1r.jar</code> file, since it conflicts with WebLogic 10.
CC-49361	If you upgrade to WebLogic 10 application server, then both ClaimCenter and ContactCenter must also be on WebLogic 10. If ClaimCenter version 4.0.8 is deployed to WebLogic 10, then Inetsoft 8 must be deployed to Weblogic 8. Also, if ClaimCenter version 4.0.8 is deployed to Tomcat 6, then Inetsoft 8 should be deployed to Tomcat 6.
PL-3793	Guidewire now supports ClaimCenter running on Tomcat 6 and WebLogic 10 with Java 1.5.
Studio	

ID	Description
CC-47005	Fixed an issue where there was a null pointer exception error in the Studio console. This occurred when you selected Verify Path in a rule sets folder and selected Verify all resources .

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, and then close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you can experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the ClaimCenter window before closing your browser.

You can optionally take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes can result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database can take longer than the default WebSphere startup timeouts allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file shows an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console. Click **Nodes**, drill down to your application server, select the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server. However, other components of the ClaimCenter environment (such as Java) can inadvertently have this requirement. If so, ClaimCenter can report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then re-start the ClaimCenter server. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Java JIT compiler options can cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's `-XX:CompileThreshold` option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this threshold to a lower value will compile methods earlier and potentially improve subsequent performance. However, the server can become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server can crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a plugin interface called `IApprovalAdapter`. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The `IApprovalAdapter` interface is not supported in ContactCenter.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short can result in duplicate work items being created. This has a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server. The default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This helps the database understand the distribution of the addresses and thus improves query performance. To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```


The database administrator should then run the commands contained in *filename*.

Out of memory errors after reloading configuration many times (CC-28736)

Issue: After reloading the configuration files many times (perhaps 10-12 times), you can get out of memory errors.

Workaround: Restart the server.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` automatically is set into the exposure. Otherwise it is left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` OR `LossParty`.

Workaround: You should add a pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaim depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: `IClaim` depends on `SynchStateData`, and `SynchStateData` uses deprecated functions like `setMessageSinkID` and `getMessageSinkID`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

List of messages returned from PendingMessages() is not ordered correctly (CC-32661)

Issue: The list of messages returned from `MessageContext.PendingMessages()` is not ordered by `SendOrder`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the **Checks** screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns and rerun the PCF Converter.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running IE 6 (or earlier) and you access the **ClaimCenter Desktop Activities** page and repeatedly click the **Activities** link, your IE memory usage increases dramatically.

Workaround: If you cannot prevent users from doing this, upgrade to IE 7.

GScript can confuse package names with variable names, causing a verification error (CC-36846)

Issue: PCF verification fails, and gives an incorrect error, if you use the same name for a variable that you have used for a package name.

Workaround: Use fully qualified package names.

Claimant bit not set correctly during upgrade from 3.1.4 (CC-38469)

Issue: Claimant bit denormalization was added to 4.0.x to make searches by claimant last name faster. This bit is not changed during an upgrade to 4.0.x.

Workaround: None.

An ISO library issue can create a first party payload from a third party exposure (CC-39126)

Issue: An error in the ISO.gs library creates a first party payload for a third party exposure.

Workaround: To correct the problem in the ISO.gs library, change this code:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||
    exposure.ExposureType == "PersonalPropertyDamage") {
    payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)
```

To:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||
    exposure.ExposureType == "PersonalPropertyDamage") {
    if (exposure.Claim.InsuredDenorm == exposure.ClaimantDenorm) {
        payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)
    } else {
        payload = isoPayloadGenerator.generateEditableInjuryPayload(exposure, null,
            exposure.Incident.Description, null)
    }
}
```

This causes the library, for a property exposure, to use the property payload for first party and the injury payload for third party.

Unchanged typelist files are marked as requiring a merge during upgrade (CC-43457)

Issue: The upgrader normally marks changed typelist files as needing to be merged. However, the upgrader may incorrectly mark end-of-line characters as having changed.

Workaround: Review the suggested merges, and ignore any that are based solely on the end-of-line character changing.

Removed hardcoding of PaymentType and NonEroding during recode and transfer (CC-43547)

Issue: When transferring or recoding a Supplemental payment to an open exposure, the PaymentType must change to Partial or Final. Hard coded logic would set the new payment to have a PaymentType of Partial, but it set the new payment's NonEroding flag (DoesNotErodeReserves) to have the same value as the original payment, which for a Supplement is NonEroding. Most Partial and Final payments are Eroding, so this behavior is counterintuitive and may not have been anticipated during implementation. However, it could have been anticipated and relied upon for a solution, so the behavior has been left unchanged for backwards compatibility.

Workaround: Now that this has been fixed and the hard coding removed, you can fix this behavior through GScript and PCF changes. Contact Guidewire Support for assistance.

The TransactionSet.setApprovingUser (usr, grp) method does not return false if user.credential.Active != true (CC-39980)

Issue: If users in a group are made inactive, then they cannot have any work assigned to them. Yet, the TransactionSet.setApprovingUser (usr, grp) method does not check the user.credential.Active value. While the method behaves as documented, you can have unexpected results.

Workaround: Check the value of `user.credential.Active` when using this method. For example:

```
if ( TransactionSet.setApprovingUser( usr, grp ) and usr.credential.Active ) {
    actions.exit()
}
```

The CheckStatusChanged event is incorrectly raised for an already requested check (CC-43902)

Issue: When a partial and final check on the same exposure are escalated on the same day, it can appear that the partial check is escalated twice. This occurs when the final check is escalated first and also escalates the partial check at the same time (when `CloseExposureAfterFinalPayment` is true). Then the partial check reaches its original turn, and the system attempts to escalate it, including running the *EventMessage* rule set. However, the second escalation fails with a `ConcurrentDataChangeException` message. No changes or messages are written to the database a second time. Upon retry, the check is correctly seen as already escalated.

Workaround: If you write logging messages in your *EventMessaging* rules, be aware of this situation if you see duplicate `CheckStatusChanged` events for a check.

Staging Table loaders Integrity checks allow an offsetting payment to be associated with a different check than the payment it offsets (CC-45725)

Issue: On a stopped or voided check, each original payment on the check is offset by another payment with a negative amount, on the same check. These two payments are related to each other by a `TransactionOffsetOnset` record that points to both payments. There is no integrity check to verify that these two payments belong to the same check. If they do not, there will be incorrect values for the check amounts and inconsistent financials after loading.

Workaround: You must correct the errors in data mapping, and ensure that each Voided or Stopped payment on a Voided or Stopped check also has an offset payment (with negative amount, in *Submitted* status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

The Exposure Closed Validation rule named *Open activities* misses an approval activity that does not have `ExposureId` set, but is related to the exposure (CC-45751)

Issue: The rule looks for open activities related to the exposure, which should prevent its closure. In the case where an approval activity is for a transaction set that carries more than one transaction, each against a different exposure, the `ExposureId` is not set on the activity. Therefore, the rule misses this open approval activity, and allows the exposure to be closed. Closing the exposure causes reserves to be zeroed out. Subsequently, if the approval activity is approved, it causes a reserve that was meant to set available reserves to zero, to cause a negative reserve balance. If this rule had caught this approval activity, it would have prevented the exposure from being closed.

Workaround: You must include an additional predicate in the rule's condition as seen in the following example:

```
exists( Activity in Exposure.Claim.Activities
where ((Activity.Exposure.ClaimOrder == Exposure.ClaimOrder
/* new predicate */ or (exists(exp in Activity.TransactionSet.Exposures where exp==exposure)))
and Activity.Status=="open" and
Activity.ActivityPattern != null and
Activity.ActivityPattern.ClosedClaimAvlble == "false")
```

Fixed amount length in the check portion is limited to six characters (CC-46460)

Issue: You can only put in a maximum of \$999.99 if using a decimal point (meaning that cents are used) or \$999,999 if there are no decimals. The `AdditionalPayeePortionInput` that renders these widgets uses a template file, where the `maxLength` of `Percentage` is set to 30 and `FixedAmount` is set to 6. Better values for these `maxLength` attributes would be 5 and 16 respectively.

Workaround: The AdditionalPayeePortionInput widget is using the static template AdditionalPayeePortion.html, located in Studio at: **config** → **web** → **templates** → **financials**. Edit the AdditionalPayeePortion.html file and modify the `maxLength="6"` for the FixedAmount input to be 16. Since this is a core file, you must track and take care of this change if you are going to upgrade to higher version before this fix is available in the base product.

An exception is generated while trying to view a ContactCenter contact in a ClaimCenter worksheet (CC-46804)

Issue: You cannot open a contact from the Document Creation wizard when integrated with ContactCenter.

Workaround: Change the ContextObjectContactPickerMenuItemSet.pcf file, on line 66 to:

```
<PickerMenuItem id="AddressBookContactSearch"
  label="displaykey.Java.UserOrContactPickerMenu.SearchAddressBook"
  action="AddressBookPickerPopup.push(requiredContactType, true, claim)"
  onPick="DocumentCreationInfo.setContextObjectValue(contextObjectName, PickedValue)"/>
```

Use the most explicit entry point for the AddressBookPickerPopup.pcf file in which you are specifying the claim and the boolean value for the `externalSearchEnabled` method.

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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 info, 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 `checkForDuplications` in the `NewClaimWizard.pcf` file as seen in the following example:

```
function checkForDuplications() {
  if (Wizard.checkForNewDuplicateClaims()) { NewClaimDuplicationsWorksheet.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="checkForDuplications()"
  ...
```

To allow **Next** to execute, you can create a new function such as:

```
function checkForDuplicationsNoWarning() {
  if (Wizard.checkForNewDuplicateClaims()) {
    NewClaimDuplicationsWorksheet.goInWorkspace(Wizard)
  }
}
```

Have the `onExit()` function of the Main Contacts wizard step call:

```
onExit="checkForDuplicationsNoWarning()"
```

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.

Errors when starting .ear file with WebLogic 10 (CC-48588 and CC-49416)

Issue: At server startup in WebLogic 10, errors are generated indicating that `GWServletTestRunner` cannot be loaded.

Workaround: Guidewire is aware of this issue, and will address it in a future release.

Guidewire ClaimCenter 4.0.9 Release Notes

Release 4.0.9.4

Copyright © 2009 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire ContactCenter, Guidewire Insurance Suite, and the Guidewire logo are trademarks 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 documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Numbers

- This release of Guidewire ClaimCenter is 4.0.9.4
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 8.0, build 20090508.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. You can contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 8.0 using the following license key:

L000-3FB-ERX-F2A734346096

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide*.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter

The following issues pertain to installing this release.

Software Updates

- If you use Oracle Enterprise Edition, you must use version 10.2.0.4.0. Oracle version 10.2.0.2.0 is no longer supported if you are upgrading to ClaimCenter 4.0.9.
- There are InetSoft issues with supporting WebLogic 10 as a report server. You can deploy the Guidewire ClaimCenter application on WebLogic 10. However, if you want to also use WebLogic for Guidewire Standard Reporting, then you must use WebLogic 8 as the report server.

Manually Upgrade the Rule Sets (CC-39604)

In 4.0.4, ClaimCenter introduced a new Transaction Pre-setup rule set. If you are upgrading to 4.0.9 from earlier versions of either 3.1.x or 4.0.x, you will not have a `ruleset.xml` file. Studio and ClaimCenter need this file to execute this rule set (even though it will start out empty, the file must at least exist). You must manually move a copy of this file to your upgraded configuration before running `upgrade.bat`:

To provide for the new Transaction Pre-setup rule set, you must manually add `ruleset.xml`:

1. Create the following new directory in your existing ClaimCenter installation: `ClaimCenter/config/resources/rules/PR/Transaction Pre-setup`.
2. In the same directory as these release notes, locate the file `ruleset.xml`.
3. Copy the `ruleset.xml` file into the new directory you created in step 1.

Best Practices for Upgrading Rules and Libraries

Since rules are now files kept in the `.../config/resources/rules` directory, the configuration upgrade process applies the same logic in evaluating your rules and the base and target configurations' rules to determine which rules to include in the upgraded configuration and which rules should be compared and merged. This is what the merge process tries to do:

- The merge process adds rules in the base configuration that do not exist in your configuration to your `...config/resources/rules` directory.
- Identical rules in both the base configuration and your configuration remain in this directory.
- When rules differ between the base configuration and your configuration, your rule remains in the `...config/resources/rules` directory, and the base rule is added to the merge directory.

Although this approach works well with other configuration files, it is not the optimal way for you to upgrade your rules, simply because you are likely to have modified most of the base rules with which you started. Thus, you will have many rules to merge with the target configuration.

The other major issue for merging rules is that rule execution order is important and is controlled by `ruleset.xml`. Unlike most other files, you should not edit or merge this file because the order of rule execution can drastically change the logic of rules. Studio should always generate this file.

Therefore, it is easier and safer to start with your own, existing rules and then selectively apply to them the changes that Guidewire has made between the base and target configurations.

To Merge Rules Efficiently

1. Make a copy of your entire set of rules (in the `...config/resources/rules` directory) before running the automatic upgrader tool.
2. Copy your old `...config/resources/rules` directory to your upgraded configuration. Now the upgraded configuration contains all of your original rules.
3. Regenerate the toolkit and build a `.war` file.
4. Point Studio to the upgraded configuration, which contains your old rules.
5. Review the rule differences between the base and target configurations to see a list of all the rule changes between the two releases.
6. Use Studio to import the updated `rules.xml` file.
7. Determine which rules to add (or modify and add) to your current rules based on the changes to the target.

Changes in this Release

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Improvements and General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 4.0.8 and ClaimCenter 4.0.9, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- GScript API
- typelists
- display keys

Base PCF File Changes

To view a report of the changes in the base PCF files, [click here](#) (requires the `readme_files` directory on your local disk).

Base Resource Changes

The following links require the `readme_files` directory on your local disk.

- To view a report of the changes in the base resources from ClaimCenter release 4.0.0 to 4.0.9, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.1 to 4.0.9, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.2 to 4.0.9, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.3 to 4.0.9, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.4 to 4.0.9, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.5 to 4.0.9, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.6 to 4.0.9, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.7 to 4.0.9, [click here](#).
- To view a report of the changes in the base resources from ClaimCenter release 4.0.8 to 4.0.9, [click here](#).

Notes on Upgrading Rules from Release 4.0.x

The upgrade to ClaimCenter 4.0.9 does not automatically merge in the changes that you have made to rules in previous 4.0.x releases. Therefore, we recommend that you preserve your customized rules and then, where desired, manually incorporate any changes that Guidewire made to the shipped rules provided with this release. To do so, use the following procedure:

1. Before upgrading, make a backup copy of your entire set of existing rules; that is, the complete `ClaimCenter/config/resources/rules` directory.
2. Run the upgrade tool.
3. Copy the `rules` directory from your backup to the upgraded `ClaimCenter/config/resources` directory. Allow the copy to overwrite any files that it needs.
4. Run Guidewire Studio. Import the rule changes from one (only one!) of the following files supplied with this ClaimCenter release:
 - If upgrading from 4.0.0, import the file: `ClaimCenter Rule Diffs 400-409 For Import.xml`
 - If upgrading from 4.0.1, import the file: `ClaimCenter Rule Diffs 401-409 For Import.xml`
 - If upgrading from 4.0.2, import the file: `ClaimCenter Rule Diffs 402-409 For Import.xml`
 - If upgrading from 4.0.3, import the file: `ClaimCenter Rule Diffs 403-409 For Import.xml`
 - If upgrading from 4.0.4, import the file: `ClaimCenter Rule Diffs 404-409 For Import.xml`
 - If upgrading from 4.0.5, import the file: `ClaimCenter Rule Diffs 405-409 For Import.xml`
 - If upgrading from 4.0.6, import the file: `ClaimCenter Rule Diffs 406-409 For Import.xml`
 - If upgrading from 4.0.7, import the file: `ClaimCenter Rule Diffs 407-409 For Import.xml`
 - If upgrading from 4.0.8, import the file: `ClaimCenter Rule Diffs 408-409 For Import.xml`

After you import the file, you will see all changed rules appear in Studio, disabled, and under parents named for the date of the import.

5. Review the rule differences detailed in the “Base Resource Changes” section of this Release Notes document.
6. If you want to incorporate any of the rule changes into your own rules, do this in Studio. You can either copy the changed section from the imported rule into your rule, or delete your rule and replace it with the imported rule. When you are finished, delete any remaining unused imported rules.

WARNING You should make the changes only within Studio. Do not attempt to modify the rules XML files directly, or to merge the diffs using a third-party utility.

Improvements and General Issues

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

ID	Description
Application Framework	
PL-1469	There are InetSoft issues with supporting WebLogic 10 as a report server. If you choose, you can deploy the Guidewire ClaimCenter application on WebLogic 10. However, if you want to also use WebLogic for Guidewire Standard Reporting, then you need to use WebLogic 8 as the report server.
PL-4311	Guidewire replaced the bfopdf.jar file to support Adobe XFA forms. However, it is still your responsibility to make the name of the form field in the template descriptor match the name of the form field in the PDF file. This name is not the simple field name from older PDF forms, but the fully qualified name. If ClaimCenter does not find the correct name, it logs the available field names.
PL-4524	Updated the PDF Library to correct a security issue for PDFs containing JavaScript.
Core	
PL-2201	Corrected an issue in which importing administration (user) data would incorrectly overwrite existing data. This occurred even if you specified <i>do not</i> overwrite data during a merge resolution.
PL-2929	Fixed an issue in which the Event Messages screen (accessed from the Administration tab) did not display the correct error message if you had more than one failed message for a claim.
PL-2930	Corrected an issue in which the <i>Claims with any unfinished messages</i> filter (in the Administration , Event Messages tab) incorrectly displayed claims that did not fall into that category.
PL-4535	Guidewire modified how it handles objects in the data object cache. Now, objects have an evict time that determines the point at which the application removes an object from the cache. This is different from the currently existing stale time value that the application uses to determine if it needs to retrieve a fresh version of an object from the database. To manage the eviction time, Guidewire added a new configuration parameter (<code>DefaultEvictTimeMinutes</code>). See the section on Cache Management in the System Administration Guide for more information.
Claim File	
CC-51274	Added a consistency check and loader validation to ensure that addresses are not shared between different claims.
Database Services	
PL-4287	Combined the following into one index: <code>EveMsgStat1</code> and <code>EveMsgError1</code> indexes and the following columns: <code>MessageSink</code> , <code>ClaimID</code> , <code>Status</code> , and <code>ErrorDescription</code> . This fix improved performance in the Event Messages screen.
Documentation and Help	
CC-49672	<p>The web service <code>IClaimFinancialsAPI</code> interface method <code>addClaimFinancialswithValidation</code> imports financials from an external system. The API also runs validation, duplicate checking, approval, and submission.</p> <p>Beginning in ClaimCenter version 4.0.5, for checks and transactions, in general set the financial transaction status for incoming financials to the <i>draft</i> status (the typecode <code>TransactionStatus.TC_draft</code>). You can also use <code>AwaitingSubmission</code> or <code>PendingApproval</code> status to mark the object for rules to transition a recently-imported check to a later status such as <i>Issued</i>. Do not set status to a <i>committed</i> status value such as <code>submitted</code>, <code>requesting</code>, <code>requested</code>, or <code>issued</code>. Those values throw exceptions on import.</p>

ID	Description
CC-52215	Guidewire added the following tables to ClaimCenter Reporting: <ul style="list-style-type: none"> cc_groupuser cc_parentgroup The table cc_state was removed.
Financials	
CC-45363	Corrected an issue where using the financials API to update a check's status to <i>cleared</i> while it was in <i>stopped</i> or <i>voided</i> status threw an exception. This status transition is now allowed.
CC-48562	Corrected an issue where approving a manual check that tried to close the exposure or claim would display an exception error. Now, ClaimCenter creates a warning activity.
CC-49383	Fixed an issue where the Financials Summary screen by claimant took too long to load on claims with a large number of claim contacts.
CC-51029	Fixed an incorrect description of a consistency check failure: Supplemental payments must have <code>doesnotorderreserves = true</code> . Previously, the description stated that supplemental payments must have <code>doesnotorderreserves = false</code> .
CC-52572	Corrected an issue where a claim-level check that was part of a bulk invoice was being escalated when its claim was closed, even though the check had <code>pendEscalationForBulk = true</code> . Now closing the claim will not escalate the check.
History	
CC-50441	When changing the litigation status from Not Litigated to Litigated in the user interface, the correct history event is now created.
ISO Integration	
CC-53395	Corrected an issue so that Medicare information could be successfully sent to ISO. The fix included correcting the code that sends the payload to ISO. This means that elements with "." and "_" can now be sent successfully.
Performance	
CC-53034	ClaimCenter 4.0.9 is supported with Oracle Enterprise Edition 10.2.0.4.0. Oracle 10.2.0.2.0 is no longer supported if you are upgrading to ClaimCenter 4.0.9.
CC-53437	Guidewire added the following configuration parameters to control whether the Team Group Activities query uses certain Oracle optimizer hints: <ul style="list-style-type: none"> DisableIndexFastFullScanForTeamGroupActivities DisableCBQTFForTeamGroupActivities DisableHashJoinForTeamGroupActivities DisableSortMergeJoinForTeamGroupActivities
Web	
CC-50711	Corrected an issue that occurred after you printed from the Dashboard tab in the user interface. The mouse pointer was stuck as an hourglass icon.
CC-51047	Fixed an error that would occur when viewing the calendar, when jumping from December 2009 to February 2010.

Known Issues and Limitations

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

Note: The maintenance release policy of Guidewire ClaimCenter is to avoid fixing configuration issues that would necessitate merging of files during a maintenance release upgrade. These issues have workarounds that can be employed directly by customers, as seen in the following section. The goal with this policy is make upgrades as straightforward as possible.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, and then close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you can experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the ClaimCenter window before closing your browser.

You can optionally take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes can result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database can take longer than the default WebSphere startup timeouts allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file shows an interruption in the “Applying upgrade step ### of #####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console. Click **Nodes**, drill down to your application server, select the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server. However, other components of the ClaimCenter environment (such as Java) can inadvertently have this requirement. If so, ClaimCenter can report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire Support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then re-start the ClaimCenter server. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin configuration in ClaimCenter/config/config.xml.

Java JIT compiler options can cause server crash on Windows with Java 1.4.2 (CC-7483)

Issue: Java's -XX:CompileThreshold option specifies how many times a method executes before the JIT compiler decides that it should be compiled. ClaimCenter uses the Java default value of 10000 for this option. Setting this

threshold to a lower value will compile methods earlier and potentially improve subsequent performance. However, the server can become overloaded by too much compilation while it is also serving ClaimCenter requests. If the threshold is too low and the server becomes overloaded, the server can crash without generating any error or log message.

Workaround: Set this threshold option to a higher value.

ContactCenter Javadoc incorrectly suggests support for IApprovalAdapter plugin interface (CC-25426)

Issue: The ContactCenter toolkit Javadoc suggests that the ContactCenter application supports a plugin interface called IApprovalAdapter. Although this is defined in ClaimCenter, this plugin is not supposed to be visible in the ContactCenter toolkit Javadoc. The IApprovalAdapter interface is not supported in ContactCenter.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short can result in duplicate work items being created. This has a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server. The default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This helps the database understand the distribution of the addresses and thus improves query performance. To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Out of memory errors after reloading configuration many times (CC-28736)

Issue: After reloading the configuration files many times (perhaps 10-12 times), you can get out of memory errors.

Workaround: Restart the server.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` automatically is set into the exposure. Otherwise it is left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaim depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: IClaim depends on SynchStateData, and SynchStateData uses deprecated functions like `setMessageSinkID` and `getMessageSinkID`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

List of messages returned from PendingMessages() is not ordered correctly (CC-32661)

Issue: The list of messages returned from `MessageContext.PendingMessages()` is not ordered by `SendOrder`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the **Checks** screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns and rerun the PCF Converter.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running IE 6 (or earlier) and you access the ClaimCenter **Desktop Activities** page and repeatedly click the **Activities** link, your IE memory usage increases dramatically.

Workaround: If you cannot prevent users from doing this, upgrade to IE 7.

GScript can confuse package names with variable names, causing a verification error (CC-36846 or PL-2377)

Issue: PCF verification fails, and gives an incorrect error, if you use the same name for a variable that you have used for a package name.

Workaround: Use fully qualified package names.

An ISO library issue can create a first party payload from a third party exposure (CC-39126)

Issue: An error in the ISO.gs library creates a first party payload for a third party exposure.

Workaround: To correct the problem in the ISO.gs library, change this code:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||
    exposure.ExposureType == "PersonalPropertyDamage") {
    payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)
```

To:

```
if (exposure.ExposureType == "PropertyDamage" || exposure.ExposureType == "LossOfUseDamage" ||
    exposure.ExposureType == "PersonalPropertyDamage") {
    if (exposure.LossParty == "insured") {
        payload = isoPayloadGenerator.generateEditablePropertyPayload(exposure, null)
    } else {
```

```

        payload = isoPayloadGenerator.generateEditableInjuryPayload(exposure, null,
                                                                    exposure.Incident.Description, null)
    }
}

```

This causes the library, for a property exposure, to use the property payload for first party and the injury payload for third party.

Unchanged typelist files are marked as requiring a merge during upgrade (CC-43457)

Issue: The upgrader normally marks changed typelist files as needing to be merged. However, the upgrader may incorrectly mark end-of-line characters as having changed.

Workaround: Review the suggested merges, and ignore any that are based solely on the end-of-line character changing.

Removed hardcoding of PaymentType and NonEroding during recode and transfer (CC-43547)

Issue: When transferring or recoding a Supplemental payment to an open exposure, the PaymentType must change to Partial or Final. Hard coded logic would set the new payment to have a PaymentType of Partial, but it set the new payment's NonEroding flag (DoesNotErodeReserves) to have the same value as the original payment, which for a Supplement is NonEroding. Most Partial and Final payments are Eroding, so this behavior is counterintuitive and may not have been anticipated during implementation. However, it could have been anticipated and relied upon for a solution, so the behavior has been left unchanged for backwards compatibility.

Workaround: Now that this has been fixed and the hard coding removed, you can fix this behavior through GScript and PCF changes. Contact Guidewire Support for assistance.

The TransactionSet.setApprovingUser (usr, grp) method does not return false if user.credential.Active != true (CC-39980)

Issue: If users in a group are made inactive, then they cannot have any work assigned to them. Yet, the TransactionSet.setApprovingUser (usr, grp) method does not check the user.credential.Active value. While the method behaves as documented, you can have unexpected results.

Workaround: Check the value of user.credential.Active when using this method. For example:

```

if (TransactionSet.setApprovingUser(usr, grp) and user.credential.Active) {
    actions.exit()
}

```

The CheckStatusChanged event is incorrectly raised for an already requested check (CC-43902)

Issue: When a partial and final check on the same exposure are escalated on the same day, it can appear that the partial check is escalated twice. This occurs when the final check is escalated first and also escalates the partial check at the same time (when CloseExposureAfterFinalPayment is true). Then the partial check reaches its original turn, and the system attempts to escalate it, including running the *EventMessage* rule set. However, the second escalation fails with a ConcurrentDataChangeException message. No changes or messages are written to the database a second time. Upon retry, the check is correctly seen as already escalated.

Workaround: If you write logging messages in your *EventMessaging* rules, be aware of this situation if you see duplicate CheckStatusChanged events for a check.

Staging Table loaders Integrity checks allow an offsetting payment to be associated with a different check than the payment it offsets (CC-45725)

Issue: On a stopped or voided check, each original payment on the check is offset by another payment with a negative amount, on the same check. These two payments are related to each other by a TransactionOffsetOnset record that points to both payments. There is no integrity check to verify that these

two payments belong to the same check. If they do not, there will be incorrect values for the check amounts and inconsistent financials after loading.

Workaround: You must correct the errors in data mapping, and ensure that each Voided or Stopped payment on a Voided or Stopped check also has an offset payment (with negative amount, in *Submitted* status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

The Exposure Closed Validation rule named *Open activities* misses an approval activity that does not have `ExposureId` set, but is related to the exposure (CC-45751)

Issue: The rule looks for open activities related to the exposure, which should prevent its closure. In the case where an approval activity is for a transaction set that carries more than one transaction, each against a different exposure, the `ExposureId` is not set on the activity. Therefore, the rule misses this open approval activity, and allows the exposure to be closed. Closing the exposure causes reserves to be zeroed out. Subsequently, if the approval activity is approved, it causes a reserve that was meant to set available reserves to zero, to cause a negative reserve balance. If this rule had caught this approval activity, it would have prevented the exposure from being closed.

Workaround: You must include an additional predicate in the rule's condition as seen in the following example:

```
exists( Activity in Exposure.Claim.Activities
where ((Activity.Exposure.ClaimOrder == Exposure.ClaimOrder
/* new predicate */ or (exists(exp in Activity.TransactionSet.Exposures where exp==exposure)))
and Activity.Status=="open" and
Activity.ActivityPattern != null and
Activity.ActivityPattern.ClosedClaimAvlble == "false")
```

Fixed amount length in the check portion is limited to six characters (CC-46460)

Issue: You can only put in a maximum of \$999.99 if using a decimal point (meaning that cents are used) or \$999,999 if there are no decimals. The `AdditionalPayeePortionInput` that renders these widgets uses a template file, where the `maxLength` of `Percentage` is set to 30 and `FixedAmount` is set to 6. Better values for these `maxLength` attributes would be 5 and 16 respectively.

Workaround: The `AdditionalPayeePortionInput` widget is using the static template `AdditionalPayeePortion.html`, located in Studio at: `config → web → templates → financials`. Edit the `AdditionalPayeePortion.html` file and modify the `maxLength="6"` for the `FixedAmount` input to be 16. Since this is a core file, you must track and take care of this change if you are going to upgrade to higher version before this fix is available in the base product.

An exception is generated while trying to view a ContactCenter contact in a ClaimCenter worksheet (CC-46804)

Issue: You cannot open a contact from the Document Creation wizard when integrated with ContactCenter.

Workaround: Change the `ContextObjectContactPickerMenuItemSet.pcf` file, on line 66 to:

```
<PickerMenuItem id="AddressBookContactSearch"
label="displaykey.Java.UserOrContactPickerMenu.SearchAddressBook"
action="AddressBookPickerPopup.push(requiredContactType, true, claim)"
onPick="DocumentCreationInfo.setContextObjectValue(contextObjectName, PickedValue)"/>
```

Use the most explicit entry point for the `AddressBookPickerPopup.pcf` file in which you are specifying the claim and the boolean value for the `externalSearchEnabled` method.

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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 info, 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.

Errors when starting .ear file with WebLogic 10 (CC-48588 and CC-49416)

Issue: At server startup in WebLogic 10, errors are generated indicating that `GWServletTestRunner` cannot be loaded.

Workaround: Guidewire is aware of this issue, and will address it in a future release.

Integrity checks allow offsetting payments to be associated to a different check than the payment it offsets (CC-45725)

Issue: Integrity checks for staging tables do not ensure that the two payments pointed to by a `TransactionOffsetOnset` row belong to the same check.

Workaround: You must correct the errors in data mapping, and ensure that each Voided or Stopped payment on a Voided or Stopped check also has an offset payment (with negative amount, in Submitted status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

Bulk invoice item processing can take a long time (CC-50253)

Issue: Bulk invoice processing performance can vary greatly. Performance depends on factors that include but are not limited to server hardware, rule set code, and server configuration settings.

Workaround: Before you deploy your final production configuration of your ClaimCenter server, you **must** test your ClaimCenter configuration using the same hardware as your production server (either the same physical hardware or an exact copy). See the *Bulk Invoice Processing and Performance* section in the ClaimCenter Integration Guide for details.

Duplicate claim search velocity template is incorrect (CC-50686)

Issue: There is a bug in the released version of the `claim.vm` file, the template used to construct the duplicate claim query used in the New Claim wizard. The query has two parts:

- The first looks for a claim with the same policy number and a loss date within three days of the current claim. This part of the query is correct.
- The second looks for a claim with the same insured and a loss date within three days of the current claim. This part of the query is incorrect.

Workaround: Find the following lines:

```
#if ($Util.getEntityIntrinsicType("Person").isAssignableFrom($Claim.insured.entityIntrinsicType))
    AND cc_contact_2.FirstNameDenorm = $Util.makeParam("Claim.Insured.Name",
        $Claim.insured.firstName)
    AND cc_contact_2.LastNameDenorm = $Util.makeParam("Claim.Insured.Name", $Claim.insured.lastName)
#else
    AND cc_contact_2.NameDenorm = $Util.makeParam("Claim.Insured.Name", $Claim.insured.name)
#end
```

Change them to:

```
#if ($Util.getEntityIntrinsicType("Person").isAssignableFrom($Claim.insured.entityIntrinsicType))
    #if ($Claim.insured.firstName)
        #set($firstNameDenorm=$Claim.insured.firstName.toLowerCase())
    #else
        #set($firstNameDenorm="")
    #end
    #if ($Claim.insured.lastName)
        #set($lastNameDenorm=$Claim.insured.lastName.toLowerCase())
    #else
        #set($lastNameDenorm="")
    #end
    AND cc_contact_2.FirstNameDenorm = $Util.makeParam("Claim.Insured.Name", $firstNameDenorm)
    AND cc_contact_2.LastNameDenorm = $Util.makeParam("Claim.Insured.Name", $lastNameDenorm)
#else
    #if ($Claim.insured.Name)
        #set($nameDenorm=$Claim.insured.Name.toLowerCase())
    #else
        #set($nameDenorm="")
    #end
    AND cc_contact_2.NameDenorm = $Util.makeParam("Claim.Insured.Name", $nameDenorm)
#end
AND cc_contact_2.Retired = 0 AND cc_claim_head.Retired = 0
#end
```

Without these changes, the second part of the query rarely matches anything on a case sensitive database.

Activity remains in queue (CC-52458)

Issue: If you delete a queue that has activities assigned to it, the activities cannot be reassigned to a user.

Workaround: You must first determine how many activities are in this state and then remove them either using a SQL query or through Gosu. Contact Guidewire Support for details on how to do this.

Bundled payments are not closing on exposure and claim (CC-50914 and CC-52374)

Issue: When a claim has two final payments using two different exposures, if one exposure is closed, the other exposure and claim are not automatically closed.

Workaround: Write a transaction validation rule to prevent this from occurring.

The AssignmentQueue list view page is uneditable (CC-50899)

Issue: In the user interface, the paging controls become disabled when you try to reassign an activity by searching for a queue.

Workaround: In Studio, open the AssignmentQueueLV file and set `editable = false` in the `ListViewPanel` properties.

Deleting a recurring check set causes an exception error (CC-53191)

Issue: When deleting a recurring check set with a check that has a status of *Requesting*, ClaimCenter throws an exception error.

Workaround: In the `ChangeCheckRecurrence.pcf` file, there is a code block at the end of the file with a function `saveCheckSetAndGo()`. Change the function to the following:

```
function saveCheckSetAndGo(){
    for (currentCheck in Check.CheckSet.Checks) {
        if (currentCheck.Editable) {
            financials.CheckUtil.saveRecurringCheckSet(Check.CheckSet);
            break
        }
    }
    CurrentLocation.commit();
}
```

Issues when purging a claim (CC-52728)

Issue: Purging a claim and running a database consistency check produces inconsistency errors.

Workaround: Guidewire is aware of this issue and recommends that you contact Guidewire Support for assistance.

Error when removing claim flags from the Team tab (CC-50160)

Issue: The code to remove the flagged status from a claim in the Team view only works with one claim at a time. The following PCFs allow you to select multiple claims in the list view and then click **Remove Flag**, which results in an error. This can happen with these PCF files: `TeamGroupOtherClaims.pcf`, `TeamGroupPendingClaims.pcf`, `TeamUserClaims.pcf`.

Workaround: Modify the flags attribute of the Remove Flag button in each `.pcf` file to read:

```
flags="one CanRemoveFlag" .
```

Guidewire ClaimCenter 5.0.0 Release Notes

Release 5.0.0.387

Copyright © 2001-2008 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 5.0.0.387

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact technical support at support@guidewire.com.

Installing This Release

For installation information, please refer to the *ClaimCenter Installation Guide*.

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

For a list of configuration and GScript API changes between ClaimCenter 4.0.5 and ClaimCenter 5.0.0, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- `gw.api.*` GScript API
- typelists
- display keys

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the `[HKEY_CURRENT_USER/Software/Microsoft/Windows/`

CurrentVersion/Explorer/BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin configuration in ClaimCenter/config/config.xml.

Using a message handling plugin to deny a check containing a final payment causes an error (CC-27165)

Issue: Using a message handling plugin to deny a check containing one or more 'final' payments causes an error. Denying checks that contain only 'partial' and 'supplemental' payments via a message handling plugin works correctly.

Workaround: Use the IClaimFinancialsAPI to deny a check with a final payment instead of a message handling plugin. Guidewire may fix this issue in a future release.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in scheduler-config.xml for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the progressInterval attribute in the <workqueue> element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Changing a coverage now fires the *Policy_Changed* event in addition to the *Coverage_Changed* event (CC-29564)

Issue: Changing a field on a Coverage now fires the *Policy_Changed* event, in addition to the *Coverage_Changed* event that has always fired.

Workaround: None.

You may not extend view entities in extension.xml with <extension> tags (CC-29717)

Issue: Unlike most entities, you may not extend a view entity, such as *RecoveryView* or *TransactionView*, with the normal mechanism of using <extension> tags in extension.xml.

Workaround: You can simply comment out the <extension> code and replace it with <viewEntityExtension> tags to extend the view entity.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the *ab_abaddress* table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Immediate updates to Change and its Total are disabled on the Set Reserves page (CC-31467)

Issue: To avoid multicurrency issues, immediate updates to the Change column and its total have been disabled on the Set Reserves page.

Workaround: Type Alt+Shift+F to post any changes to the server, which will then update the Change column and its total.

Lists of claimants are not the same in different PCFs (CC-31484)

Issue: *ClaimSearchResultsLV.pcf* uses a method to display claimants that concatenates claimants in a comma delimited list. Such a list is not correctly internationalized and could be very long. In other list views, such as *DuplicateClaimSearchResultsLV.pcf*, *Claim.Claimant* values are shown. This list does not take the Exposure level claimants into account (an auto claim ends up having no claimant in this PCF). The net effect is that lists of claimants can differ inside ClaimCenter.

Workaround: None; be aware that lists of claimants can vary, depending on how they are generated. You can make these lists a little more similar by setting the *MaxClaimantsInClaimListsViews* configuration parameter to a small value. Do not set this to zero (the default) because then no limit is imposed.

Foreign exchange adjustments trip aggregate limit warnings (CC-31517)

Issue: When a foreign exchange adjustment is applied to a payment, it could cause aggregate limits to be reached and trigger a validation warning or error. This is not correct behavior, since the adjustment could not make a significant change to the amount.

Workaround: None.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` method in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a Pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

Bulk invoices created with the bulk invoice API will not go from Awaiting Submission to Requesting when `bulkinvoicesescalation` is executed (CC-31567)

Issue: The `bulkinvoicesescalation` batch process correctly escalates API-created bulk invoices on the correct date. However, if the time of the invoice is late, the batch process might miss the invoice until the next day.

Workaround: Set the `ScheduledSendDate` of the batch invoice to `null`.

`IClaimAPI` depends on `SynchStateData`, which uses deprecated methods (CC-31611)

Issue: `IClaimAPI` depends on `SynchStateData`, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

Foreign exchange adjustments should not apply to some payment amounts (CC-31882)

Issue: When an exchange adjustment is applied to a check, the adjustment is split proportionally across all the payments on the check. This split should, but does not, exclude payments that do not actively contribute to the gross amount of the check, on which the adjustment is made. This includes any voided, stopped, recoded, or transferred payments, and any offsets related to such payments.

Workaround: None.

An exception is thrown when trying to link a new contact to the address book from the new check wizard (CC-32732)

Issue: If you create a new contact in the New Check Wizard, and later attempt to link contacts to `ContactCenter`, you get an error.

Workaround: None.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the `Checks` screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To get the payment's send date, both the Transactions and the Payments table would have to be searched; this would hamper performance and is not done.

Workaround: None.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns, then rerun the PCF Converter.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The Reset button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the Reset button an Initialize button.

SQL inline parametric query limit of 2100 gives an error when there are >2100 users in a group (CC-36172)

Issue: If you generate a query that exceeds the SQL inline parametric query limit of 2100, you will receive an error. For example, if you supervise many groups with many users, your checks of claim ACLs can generate a query that reaches this limit.

Workaround: None.

After logout, a request to return to the login page creates a new session in Websphere (CC-36394)

Issue: On logout in WebSphere, the current session is destroyed but a new one is created. In ClaimCenter, logging out takes you back to the login page. Whenever a browser interacts with an application server, the server creates a session if none existed.

Workaround: On logout, go outside of the Guidewire application and enforce static content caching: the browser will reuse static objects in its cache (like the login page) and therefore minimize the unnecessary interactions with the application server. Alternately, have a web tier cache the static content. This is basically adding another caching layer (on top of the browser) and would reduce this phenomenon further.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you go to the ClaimCenter **Desktop Activities** page and repeatedly click **Activities**, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to Internet Explorer 7.

The Metro order date is initially incorrect (CC-37168)

Issue: In the New Claim wizard, the initial **Order Date** shown when ordering a Metro police report incorrectly contains the date the report request was created. The date should be blank - no report has been requested yet.

Workaround: Correct this date while ordering the report.

Archiving a claim with aggregate limits can cause an issue with the aggregate limits (CC-37207)

Issue: When a claim is assigned to a different policy, the aggregate limits of the new (and/or previous) PolicyPeriod are marked as invalid. The aggregate limits are then recalculated by looking at all contributing claims. This cannot be done when one of the claims contributing to the aggregate limit has been archived, and the aggregate limit remains incorrect.

Workaround: Write a rule in the Archive Rule Set to prevent claims with aggregate limits from being archived.

The configuration upgrade tool can cause an error when moving rules (CC-37988)

Issue: The ClaimCenter 5.0 configuration upgrade tool moves the entire rule set of the customer configuration into the proper module/configuration/... location in the working directory's customer configuration. However, if the customer configuration directory lacks a /rules folder, the tool throws an exception and stops the upgrade process.

Workaround: Make sure that the rules folder exists in the customer configuration (config/resources/rules folder) exists before running the upgrade tool.

If changed, you must manually merge WC-prefixed typelists during an upgrade to 5.0.0 (CC-38166)

Issue: Both ClaimCenter 3.1.x and 4.0.x contain five pairs of typelists related to injury incidents; see the following table. In version 5.0.0, the WC- prefixed typelists are no longer used by the default implementation, and the un-prefixed typelists now contain the typecodes of both. Fields on Exposure that had used the WC- prefixed typecodes now use the typecodes from the corresponding un-prefixed typelists, and these Exposure fields were moved to InjuryIncident.

If you have modified any of these WC-prefixed typelists, you must manually merge their changes and additions into the corresponding retained typelist during your configuration upgrade to 5.0.0. If you do not perform this manual merge before upgrading your database, a database upgrade Version Check will fail.

Un-prefixed typelist name	WC-prefixed typelist name
InjuryType	WCInjuryType
DetailedInjuryType	WCDetailedInjuryType
BodyPartType	WCBodyPartType
DetailedBodyPartType	WCDetailedBodyPartType
MedicalTreatmentType	WCMedicalTreatmentType

Workaround: For each WC-prefixed typelist that has been modified, you must transfer its changed and added typekeys to the un-prefixed typelist as follows:

1. Use Studio's typelist editor to copy all typekeys unique to (added to) the WC-prefixed typelist to the un-prefixed typelist.
2. For typekeys that have the same typecode in both typelists, make sure that each name in the non-prefixed typelist matches the name in the WC-prefixed typelist.

Note: Names are not case-sensitive, and the "Eyes(s)" and "Nose(s)" differences in WCDetailedBodyPartType are ignored. You can leave those spelled correctly in DetailedBodyPartType).
3. When finished, Studio copies the un-prefixed typelist to the configuration module in the working directory.

A good time to do this is while using the merge tool to merge your other typelists.

Security warning caused by installing *DocumentAssistant* ActiveX control (CC-40140)

Issue: The signature of the current version of this ActiveX control has expired; adding or using this control gives a security warning: “The driver software you are installing has not been properly signed with Authenticode (TM) technology... The publisher’s identity cannot be verified... Do you still want to install this driver software?”

Workaround: Click Yes to ignore this warning when it appears.

Round-robin assignment state is updated even if assignment is not saved (CC-40197)

Issue: If assignment rules fail to assign to a group because validation rules prevent the assignment, the assignment state updates nevertheless. This is true for assigning claims, but may also be an issue in assigning other work items.

Workaround: None.

External Editors are not recognizing Tool / Option Settings in Studio (CC-40225)

Issue: External editors that are set in the Tools / Option menu are not being recognized by the Studio application when you click on `config.xml` or `profile.properties`. Also, opening some files with an external editor gives a message asking whether to edit the file and place a copy of the file in the configuration module. A No answer still opens the file in the external editor.

Workaround: Copy the file to the configuration module and edit it there directly with the external editor.

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the Plan of Action > New Evaluation page will display an incorrect title (“New Evaluation” instead of “LOBName Evaluation”), and the page will not have a “LOBName Cost” field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names. The ‘cargo’ LOB displays these same errors.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use `ClaimEvaluationDetail` page instead of `ClaimEvaluationDetailsCargo` page. Also, do not customize the `ClaimEvaluationDetail` page heavily.

Dashboard drilldown report is broken when the group number > 300 (CC-40284)

Issue: You do not get a detailed (drilldown) report of the Dashboard Tab’s statistics when there are a large number of groups (>300) in the installation. This occurs in both Microsoft MS-SQL and Oracle databases.

Workaround: None.

Issue exists in creating some documents from a document template (CC-40425)

Issue: Document creation will fail if the document template uses any one of the following variables: Document, Claim, or DocumentCreationInfo.

Workaround: You don't need to change any template. Instead, in the action for the CreateDocument button in `NewTemplateDocumentDV.pcf`, replace this method call:

```
gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo)
```

with this call:

```
getDocumentCreationParameters()
```

and implement this method in the PCF as:

```
function getDocumentCreationParameters() : java.util.Map{
    var map = gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo)
    map.put( "Document", Document)
    map.put( "DocumentCreationInfo", DocumentCreationInfo)
    map.put( "Claim", Claim)
```

```
    return map  
}
```

Note: The context for document template processing derives from the PCF which initiates the document creation. Thus, do not remove or change the variables in `NewTemplateDocumentDV.pcf`.

Acrobat Sample document template does not allow extra fields (CC-40462)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Archiving fails on databases upgraded from version 3.1.x (CC-40464)

Issue: Upgrading 3.1.x databases introduces system data that are not compatible with archiving; the Archive batch process fails if you later attempt to archive a claim from this database.

Workaround: None.

MaxBrowserHistoryItems gives an error (CC-40466)

Issue: In version 4.0.x, you could disable the Back button by changing the `MaxBrowserHistoryItems` parameter to 0. This is no longer possible in version 5.0.0.

Workaround: None.

Upgrading does not move jars to the plugin directory (CC-40467)

Issue: The upgrade process creates a `modules/configuration/config/plugin` folder into which jar files are moved by the automatic upgrade tool. The upgrade tool expects the jar files to be in a different location (`modules/configuration/plugin`).

Workaround: Manually move the plugin directory up one level - from `modules/configuration/config/plugin` to `modules/configuration/plugin`.

The Claim Summary row of *ClaimSearchResults* has been removed (CC-40476)

Issue: The summary row has been removed from `ClaimSearchResultsLV.pcf`.

Workaround: If you would like to add back this summary row, contact Guidewire Support. Note that this can reduce performance.

FNOL wizard's Basic Info page has incorrect *post on change* behaviors (CC-40479)

Issue: On the **Basic Info** (second) page of the New Claim wizard, editing the **Reported By** contact and then switching to another contact can sometimes edit the wrong contact and can also commit multiple contact changes.

Workaround: None.

Studio fails after multiple Verify actions (CC-40499)

Issue: A memory leak during the Verify action of Studio causes Studio to use large amounts of memory; two Verify All commands result in an out-of-memory error.

Workaround: In Studio, avoid verifying everything; verify only those items you must. You may also exit and restart Studio to recover the leaked memory.

Editing a renamed rule causes a Java error (CC-40505)

Issue: Renaming and then editing a rule in Studio gives this error: “java.lang.RuntimeException: org.xml.sax.SAXParseException: Premature end of file.”

Workaround: Do not rename rules. Write new rules with the name you wish, then copy GScript into your new rule from the old one, and mark the old rule invalid. Do not edit a renamed rule outside of Studio.

IClaimFinancialAPI does not handle archived claims correctly (CC-40514)

Issue: If you try to add transactions, or otherwise try to modify, an archived claim with this API, it returns a `BadIdentifierException` error message, and does not modify the claim. You may not modify archived claims, even with an API.

Workaround: Use `IClaimAPI` to determine if a Claim is archived before attempting to use `IClaimFinancialsAPI` to add financials. You can also use `IClaimAPI` to restore the claim; this enables you to edit it.

Subrogation: On Financials tab, nulls lead to non-display of some amounts (CC-40522)

Issue: Classes such as `FinancialsCalculationUtil.getTotalRecoveries().getAmount()` return null in certain cases. In the Subrogation screens, on the Financials tab, this leads to non-display of these amounts because one component of their calculation is null.

Workaround: Edit these GScript classes so that they return zero instead of null.

You cannot delete a workflow, and an invalid workflow prevents the server from starting (CC-40533)

Issue: It is not possible to delete a workflow. Further, if you invalidate a workflow in the process of editing it, the invalidated workflow will prevent the application server from (re)starting.

Workaround: Instead of deleting an obsolete workflow, make sure that your application never triggers it. Also, use Studio’s workflow editor to make sure all workflows are valid.

The provided JMS transport plugin’s *Plugin Directory* field must be *messaging* (CC-40581)

Issue: If you enable the JMS transport plugin provided in the default implementation, an error results.

Workaround: Set its `Plugin Directory` value to `messaging`.

Studio does not move resources properly (CC-40654)

Issue: In Studio (customer mode), if you create a new package, and then create a class in this package, attempting to move this class creates several empty folders and moves the resource incorrectly.

Workaround: None.

ClaimCenter uses an old version of Castor libraries (CC-40659)

Issue: Guidewire products still use an old version of the Castor library, and you cannot modify the version used. The version in ClaimCenter is bundled version 0.9.4.3. The current version version of the Castor libraries is 1.2.

Workaround: You may use another version of this library within your plugin code. To use a different version of Castor, add its jar to the `modules/configuration/plugins/<plugindir>/lib` folder. All jar files located in the plugins directory are loaded with a different class loader than the one used by the ClaimCenter core code. This avoids any potential conflicts between versions of the libraries.

Guidewire ClaimCenter 5.0.1 Release Notes

Release 5.0.1.53

Copyright © 2008 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire ContactCenter, Guidewire Insurance Suite, and the Guidewire logo are trademarks 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 documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations
- Documentation-Related Issues

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 5.0.1.53.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact Guidewire Customer Support. You may contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For installation information, please refer to the *ClaimCenter Installation Guide*.

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Improvements
- General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 5.0.0 and ClaimCenter 5.0.1, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- `gw.api.*` GScript API
- typelists
- display keys

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.0 to 5.0.1

- 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/platform` directory, [click here](#).

Base Resource Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.0 to 5.0.1

- To view a report of the changes in the base resources in the modules/cc directory, [click here](#).
- To view a report of the changes in the base resources in the modules/core directory, [click here](#).
- To view a report of the changes in the base resources in the modules/platform directory, [click here](#).

getFieldValue/setFieldValue Changes (CC-42508)

The rarely-used entity methods `getFieldValue()` and `setFieldValue()` are now deprecated. Any code using them should be changed to use a more standard entity field access approach.

For example, do **not** use this GScript syntax:

```
var x = myEntity.getFieldValue("MyFieldName") // get
myEntity.setFieldValue("MyFieldName", "my new value for this field") // set
```

Instead, you should almost always use this type-safe GScript syntax:

```
var x = myEntity.MyField // get
x.MyField = "my new value for this field" // set
```

If necessary, you can use this alternative syntax if all you have at run time is a `String` representation of the field or property name:

```
var x = entity["MyField"] // get
entity["MyField"] = "my new value for this field" //set
```

Use of the deprecated `getFieldValue()` and `setFieldValue()` will be unsupported completely in a future release and you should begin to cover any usages of them.

However, there are certain usages that are fully **unsupported** even in the current release because they do not always return the same result as standard field access. You must search your code for and immediately convert all of the following usages:

	Property Access	Field Name
Activity	setFieldValue	AssignedGroup
Activity	setFieldValue	AssignedUser
Activity	setFieldValue	EscalationDate
Activity	setFieldValue	TargetDate
AggregateLimitRpt	getFieldValue	LimitUsed
Answer	setFieldValue	BooleanAnswer
Answer	setFieldValue	ChoiceAnswer
Answer	setFieldValue	DateAnswer
Answer	setFieldValue	IntegerAnswer
Answer	setFieldValue	TextAnswer
Assignable	setFieldValue	AssignedGroup
Assignable	setFieldValue	AssignedQueue
Assignable	setFieldValue	AssignedUser
BulkInvoice	getFieldValue	ApprovedReportingAmount
BulkInvoice	getFieldValue	ApprovedTransactionAmount
BulkInvoice	getFieldValue	BulkInvoiceTotal
BulkInvoice	setFieldValue	BulkInvoiceTotal
BulkInvoice	setFieldValue	CheckNumber
BulkInvoice	setFieldValue	Currency
BulkInvoice	getFieldValue	SueDate
BulkInvoice	setFieldValue	ScheduledSendDate
BulkInvoice	setFieldValue	Status
BulkInvoice	getFieldValue	TotalReportingAmount

	Property Access	Field Name
BulkInvoice	getFieldValue	TotalTransactionAmount
BulkInvoiceItem	getFieldValue	Amount
BulkInvoiceItem	setFieldValue	Amount
BulkInvoiceItem	getFieldValue	ClaimNumber
BulkInvoiceItem	getFieldValue	DeductionsAmount
BulkInvoiceItem	setFieldValue	NonEroding
BulkInvoiceItem	setFieldValue	PaymentType
BulkInvoiceSearchView	getFieldValue	ApprovedReportingAmount
BulkInvoiceSearchView	getFieldValue	ApprovedTransactionAmount
BulkInvoiceSearchView	getFieldValue	BulkInvoiceTotal
BulkInvoiceSearchView	getFieldValue	TotalReportingAmount
BulkInvoiceSearchView	getFieldValue	TotalTransactionAmount
Check	getFieldValue	PendEscalationForBulk
Check	getFieldValue	ReportableAmount
Check	setFieldValue	Status
CheckPortion	setFieldValue	FixedAmount
CheckPortion	setFieldValue	Percentage
Claim	setFieldValue	AssignedGroup
Claim	setFieldValue	AssignedUser
Claim	setFieldValue	ClaimNumber
Claim	setFieldValue	LitigationStatus
Claim	getFieldValue	Notes
Claim	setFieldValue	Policy
Claim	setFieldValue	PublicID
Claim	getFieldValue	RoleAssignments
ClaimAssociation	getFieldValue	ClaimsInAssoc
ClaimContactRole	setFieldValue	Evaluation
ClaimContactRole	setFieldValue	Exposure
ClaimContactRole	setFieldValue	Incident
ClaimContactRole	setFieldValue	Matter
ClaimContactRole	setFieldValue	Negotiation
ClaimContactRole	setFieldValue	Policy
ClaimInAssociation	setFieldValue	ClaimInfo
ClaimInfoSearchView	getFieldValue	ClaimStateOrder
ClaimUserModelSet	setFieldValue	Claim
Credential	setFieldValue	Password
DashboardStats	getFieldValue	ClaimCostsInPeriod
DashboardStats	getFieldValue	ClaimCostsOnClosed
DashboardStats	getFieldValue	ClaimCostsPaid
DashboardStats	setFieldValue	CloseTime
DashboardStats	getFieldValue	ClosedClaims
DashboardStats	getFieldValue	ClosedExposures
DashboardStats	getFieldValue	ExpensesInPeriod
DashboardStats	getFieldValue	ExpensesOnClosed
DashboardStats	getFieldValue	ExpensesPaid
DashboardStats	getFieldValue	Flagged
DashboardStats	getFieldValue	Handlers
DashboardStats	getFieldValue	Litigated

	Property Access	Field Name
DashboardStats	getFieldValue	NewClaims
DashboardStats	getFieldValue	NewExposures
DashboardStats	getFieldValue	NewLitigation
DashboardStats	getFieldValue	NewNoticeOnly
DashboardStats	getFieldValue	NoticeOnly
DashboardStats	getFieldValue	OpenClaims
DashboardStats	getFieldValue	OpenExposures
DashboardStats	getFieldValue	OpenReserves
DashboardStats	getFieldValue	OverIncurredLimit
DashboardStats	getFieldValue	RecoveredInPeriod
DashboardStats	getFieldValue	ReopenedClaims
DashboardStats	getFieldValue	TotalIncurredNet
DashboardStats	getFieldValue	TtlIncNetMinusOpenRecReserves
Deduction	getFieldValue	Amount
Deduction	setFieldValue	Amount
Evaluation	setFieldValue	CreateTime
Exposure	setFieldValue	AssignedGroup
Exposure	setFieldValue	AssignedUser
Exposure	setFieldValue	ExposureType
Exposure	setFieldValue	Incident
GroupUser	setFieldValue	LoadFactor
Holiday	setFieldValue	AppliesToAllZones
LitStatusTypeLine	setFieldValue	CompletionDate
LocationBasedRU	setFieldValue	PolicyLocation
NameCriteria	setFieldValue	Name
Negotiation	setFieldValue	CreateTime
Note	setFieldValue	Body
Payment	setFieldValue	PaymentType
Region	getFieldValue	RegionType
Review	setFieldValue	Claim
Review	setFieldValue	ReviewType
RiskUnit	getFieldValue	extractready
RiskUnit	setFieldValue	extractready
SREEReport	setFieldValue	Description
TAccountLineItem	setFieldValue	CreditingTransaction
TAccountLineItem	setFieldValue	DebitingTransaction
Transaction	setFieldValue	Currency
Transaction	setFieldValue	ReserveLine
TransactionEditWrapper	setFieldValue	NewAmount
TransactionLineItem	getFieldValue	ClaimAmount
TransactionLineItem	setFieldValue	ClaimAmount
TransactionLineItem	getFieldValue	ClaimForExAmount
TransactionLineItem	getFieldValue	ReportingAmount
TransactionLineItem	getFieldValue	ReportingForExAmount
TransactionLineItem	setFieldValue	Transaction
TransactionLineItem	getFieldValue	TransactionAmount
TransactionLineItem	setFieldValue	TransactionAmount
VehicleRU	setFieldValue	VehicleLocation

	Property Access	Field Name
Workflow	setFieldValue	timeoutTime

Before your next upgrade, ensure you remove unsupported usages of entity methods set/get field value to instead use standard field value accesses in GScript.

Before starting the server, you must remove the explicitly unsupported usages of entity methods set/get field value to instead use standard field value accesses in GScript. Refer to the reference table in this section for details.

TypeInfo Subobjects Changed Package (CC-41379)

The package hierarchies changed for some objects that hang off of the `TypeInfo` class. These objects would only be used for advanced language reflection (run time analysis of type information or object introspection). Type information classes, such as those that encapsulate methods on a class, that were defined in the `com.guidewire.*` hierarchy are now defined in the `gw.lang.reflect.*` hierarchy.

New TypeSystem Class (CC-41379)

You can use the new class `gw.lang.reflect.TypeSystem` for additional supported APIs for advanced type system introspection. For example, its `getByFullName` method can return a `Type` object from a `String` containing its fully-qualified name.

For example, the following code gets a type by a `String` version of its fully-qualified name and instantiates it using the type information for the type:

```
var myFullClassName = "com.mycompany.MyType"
var type = TypeSystem.getByFullName( myFullClassName )
var instance = type.TypeInfo.getConstructor( null ).Constructor.newInstance( null )
```

Refer to the GScript API Reference in Studio in the **Help** menu for details of additional methods on this class.

Changes to Integration Entity Libraries in 5.0.1 (CC-41164)

In earlier releases of ClaimCenter, GScript library support was deprecated and replaced with code implemented as GScript classes (for general utilities) or GScript enhancements (for extensions methods to be added to entities). As a consequence of this change, Java code implementing Java plugins or other Java called from GScript could not call Java libraries or access static methods and static fields on Java classes.

ClaimCenter 5.0.1 adds additional features that make it easier to call GScript utilities from Java. This is especially relevant for customers upgrading from earlier versions of ClaimCenter that used the `invokeLibraryMethod` from Java to call *libraries*, which no longer exist as of version ClaimCenter 5.0.0.

GScript Classes Now Exported to Java

GScript classes and instances of GScript classes are now exported to Java through the entity libraries. For example, a GScript class called `MyUtils` will be exported into Java into the package `external.*` as a class called `MyUtils`. All methods and properties are dynamically translated from GScript to Java, and back again as needed. If you want to change the package for the class, refer to later in this section for package mapping configuration.

To construct a new instance of a GScript class, you can use an exposed static object in a field called `CONSTRUCT` that can be referenced from the class. It contains methods that mirror the GScript constructors.

For example, to create an instance of a GScript class `MyClass` from Java using an empty constructor, use the code:

```
MyClass myInstance = MyClass.CONSTRUCT.newInstance();
```

If there are other constructors (in other words, constructors that take arguments), use the alternate exposed versions of the `newInstance()` method that have the same types of arguments. For example, to create an instance of a GScript class `MyClass` from Java using a constructor that takes a `String` argument, use the syntax:

```
MyClass myInstance = MyClass.CONSTRUCT.newInstance("My argument to the constructor");
```

This is a new feature, and there is no required upgrade step associated with this change.

GScript Enhancements Now Exported to Java

There is now support for exporting GScript enhancements to Java code through the entity libraries, which are the generated Java libraries used to access the Guidewire entities. Refer to the GScript Reference Guide in the “Enhancements” chapter for more information about GScript enhancements. Both properties and methods defined in enhancements are exported to the Java world. There is no special task necessary to use this feature, the enhanced properties and methods are simply available on the Guidewire entities after you regenerate the entity libraries, which is part of regenerating the toolkit.

This is a new feature, and there is no required upgrade step associated with this change.

Static Methods and Fields Exported Using UTIL Field

When referenced through the Java entity libraries, static methods and static fields on Java classes and GScript classes are exposed on classes, although not directly on the class. Static methods and static objects are encapsulated in a new anonymous class referenced through the UTIL field of each class.

For example, suppose a Java class had a static field like this:

```
String MYSTATICFIELD = "Hello";
```

It is now exposed to Java through the entity libraries as an anonymous class in the UTIL field:

```
static class UTIL {  
    ...  
    public static String getMYSTATICFIELD() {  
        ...  
    }  
}
```

For the small number of Guidewire Java classes that exposed static fields or methods, this is a change. Before starting the server, ensure you change any affected code that flagged as a syntax error.

To show how this affects your own code, any Java code that had this form:

```
String myStaticValue = MyClass.MYSTATICFIELD;
```

...must be replaced with code of the form:

```
String myStaticValue = MyClass.UTIL.getMYSTATICFIELD();
```

You can read static GScript properties on Java classes (using the *enhancements* feature to add methods or properties) or static GScript properties on native GScript classes using code like:

```
String geocodeStatus = MyClass.UTIL.getMYSTATICPROPERTY();
```

Similarly, you can set static GScript enhancement properties with code like:

```
String geocodeStatus = MyClass.UTIL.setMYSTATICPROPERTY("new value here");
```

For more information about GScript classes, the `static` modifier, enhancements, and GScript properties, refer to the GScript Reference Guide.

External Entities Packages Unchanged in This Release, But Now Configurable

In earlier releases, external entity classes always had the namespace of:

```
com.guidewire.cc.external.entity
```

Typelists had the namespace of

```
com.guidewire.cc.external.typelist
```

In this release, these namespaces are configurable in a file in the configuration called `ExternalMapping.properties`.

```
.../modules/configuration/config/plugin/ExternalMappings.properties
```

This file maps internal namespaces to external namespaces. To ensure backwards compatibility, the included mapping for entities is preserved.

The format for this properties file for namespaces is as follows:

```
INTERNAL_GSCRIPT_NAMESPACE. = EXTERNAL_GSCRIPT_NAMESPACE
```

Note the period after the internal GScript namespace identifier.

If you wish to map a single type, then simply list the fully qualified name before and after for that type. e.g.:

```
INTERNAL_TYPE_FULLYQUALIFIEDNAME = EXTERNAL_TYPE_FULLYQUALIFIEDNAME
```

For example:

```
gw.plugin.addressbook.IAddressBookAdapter = com.guidewire.ab.plugin.addressbook.IAddressBookAdapter
gw.plugin.document.IDocumentContentSource = com.guidewire.ab.plugin.document.IDocumentContentSource
```

The previous behavior of packages is preserved because it contains one entry:

```
entity. = com.guidewire.cc.external.entity
```

This is a new feature, and there is no required upgrade step associated with this change.

Additional Java Types Exported to Java Entity Libraries

In addition to GScript types exported to Java, the following Java types accessed in GScript are now exported to Java in the entity libraries in the following cases.

This list did not change in this release, but is published for clarity regarding what classes are exported:

Type of class	Exposed	Translated with intermediate proxy class
Built-in Java classes related to plugin interface method parameters or return types, or other critical integration tasks	Yes	Yes
Java classes in the <code>java.*</code> namespace	Yes	No
Java classes in the <code>javax.*</code> namespace	Yes	No
Additional classes exposed in the libraries:	Yes	No
• <code>gw-plugin.jar</code>		
• <code>gw-plugin-cc.jar</code>		

This is a new feature, and there is no required upgrade step associated with this change.

Messaging Plugins and IBaseURLBuilder Plugin Changes (CC-42481)

Starting in this release, implementation messaging plugins and the `IBaseURLBuilder` plugin must explicitly implement `InitializablePlugin` in the class definition. This interface tells the application that you support the `setParameters` method to get parameters from the plugin registry. If you do not already have this defined explicitly, you must also implement `InitializablePlugin` or the application will not initialize your plugin if it is a messaging plugin or a base URL builder plugin implementation.

For example, suppose your plugin implementation's first line looks like this:

```
class FooTransport implements MessageTransport {
```

Change it to this:

```
class FooTransport implements MessageTransport, InitializablePlugin {
```

To conform to the new interface, you must also implement a `setParameters` method even if you do not need parameters from the plugin registry:

```
function setParameters(map: java.util.Map) { // this is part of InitializablePlugin
```

```
// here we can access values in the MAP to get parameters defined in plugin registry in Studio
}
```

Geocoding Plugin Changes (CC-39806)

There are two geocoding plugin interfaces, and both changed name in this release.

You must update any references to these interfaces in your code.

Old Name	New Name	Description
GeocodePlugin	IGeocodePlugin50	This is the new geocoding interface introduced in release 5.0.0
IGeocodePlugin	IGeocodePlugin40	This is the deprecated geocoding interface from release 4.0.X

Changes in Java for IGeocodePlugin40

In addition to the name change mentioned earlier, due to changes to the Java entity libraries, if you use the Java version of the ClaimCenter 4.0.X included plugin implementation of IGeocodePlugin40, you may need to make some minor changes. Note that IGeocodePlugin is the ClaimCenter 4.0.X (older) plugin interface, in contrast to the newer GeocodePlugin interface introduced in ClaimCenter 5.0.0.

The IGeocodePlugin interface used to have this field declaration:

```
String GEOCODE_STATUS = "GeocodeStatus";
```

It has been replaced with

```
static class UTIL {
    ...
    public static String getGEOCODE_STATUS() {
        ...
    }
}
```

Because of this, Java code of this form:

```
String geocodeStatus = IGeocodePlugin40.GEOCODE_STATUS;
```

...must be replaced with code of the form:

```
String geocodeStatus = IGeocodePlugin40.UTIL.getGEOCODE_STATUS();
```

If you use the Java version of the old geocoding plugin interface, you must these changes in your Java code.

Changing Coverage Triggers Policy Changed Event (CC-29564)

Due to changes to the policy model in ClaimCenter, changing a field on a Coverage entity now fires the PolicyChanged event, in addition to the CoverageChanged event that has always fired. Review your Event Fired rule set code to ensure it expects changing a field on a coverage to also fire the policy changed event.

Advanced Authentication to ContactCenter

Typical authentication from ClaimCenter to ContactCenter uses the name and password defined in the IAddressBookAdapter plugin in Studio for the ClaimCenter application. Starting in ClaimCenter 5.0.1, if you want to hide authentication information in a separate file, you can implement a special plugin interface specifically for this purpose, called ABAuthenticationPlugin. This is similar to the use of DBAuthentication plugin to connect to ContactCenter that was used in ClaimCenter 4.0.x. Refer to “ABAuthenticationPlugin for ContactCenter Authentication” in the *Integration Guide* for details.

Improvements

The following are other improvements made to ClaimCenter for this release:

Issue ID	Description
CC-31484	<p>Claimant list for claims in LVs now are all displayed using ContentCell/ListIterators rather than synthesized in Java code. In addition to the aforementioned LVs, the Claimant lists for Claims listed in the AddressBook search, Team Claims, Vacation Claims, Archived Claims and Simple Claim search results are also affected by this change.</p> <p>Guidewire has added a new configuration parameter: <code>MaxClaimantsInClaimListViews</code>. This new parameter specifies the maximum number of claimants that should be listed for each claim in a LV. If set to zero (the default) then no limit is imposed.</p>
CC-31790	Modified search for duplicate checks in the Check wizard and <code>IClaimFinancialsAPI</code> to include a restriction for the currency of the check being matched. This prevents a 50 Euro check from incorrectly matching a \$50 check.
CC-32909	The deprecated <code>Exposure.Transactions</code> has been replaced by <code>Exposure.getTransactionsIterator(false)</code> in the <code>Exposure</code> library class extensions <code>totalForCoverageType()</code> and <code>totalForCoverageTypeandClaimant()</code> .
CC-35518	The <code>StatCode</code> typelist is now sorted when displayed on the policy Statistical Data page.
CC-35954	<p>You can no longer create aggregate limit on a policy period with archived claims. ClaimCenter displays an error message stating "An aggregate limit cannot be on a policy period with archived claims. Restore the following claim(s) before creating this aggregate limit: [archived claim number(s)]".</p> <p>You can, however, create an aggregate limit if archived claims are on the same policy but on a different policy period.</p>
CC-37168	The New Claim wizard Loss Details page now displays the "sent" date for the Metro Report, rather than the "created" date.
CC-38231	Many text strings that appeared within sample rules have been replaced by display keys.
CC-38765	<p>Added the following methods:</p> <ul style="list-style-type: none"> <code>Check.getClaimForExAdjustmentAmount()</code> <code>Check.getReportingForExAdjustmentAmount()</code>
CC-39072	Multiple deferred requests for a metro report no longer create multiple activities.
CC-39323	Restricted archive search to only return a result link for those users who have the required permission to view archived claims.
CC-39827	<p>Added the following new methods to <code>gw.api.util.Logger</code> that you can use to conditionally construct debug strings based on the logging level:</p> <ul style="list-style-type: none"> <code>isDebugEnabled()</code> <code>isTraceEnabled()</code> <code>isInfoEnabled()</code>
CC-39943	Added a consistency check that <code>FixedClaimAmount</code> IS NOT NULL if the Check is multicurrency. Also, added a Loader integrity check to verify that <code>CheckPortion.FixedClaimAmount</code> is populated. (This value must be either the same as <code>FixedAmount</code> , or different in the case of multicurrency multi-portion checks.)
CC-40005	Modified the Claim and Exposure validation rules to only validate claim and exposure roles if something relevant has changed (for example, if a <code>ClaimContact</code> or a <code>ClaimContactRole</code> has changed).
CC-40139	Modified <code>CheckSet.ExceedsAvailableReserves()</code> to handle calculations for Recurrence checks. If the <code>CheckSet</code> has a recurrence, it needs to multiply all the payment amounts by the number of checks, so the total of the recurrence is taken into account (but without creating the recurring checks).
CC-40162	<p>Made available domain methods that include <code>CurrencyAmount</code> in their signature on the external entities. (This is useful for Java plugins). <code>TransactionLineItem.TransactionAmount</code>, and <code>CurrencyAmount</code> have been hidden again as intended, and code should now use the following:</p> <ul style="list-style-type: none"> <code>TransactionLineItem.getTransactionCurrencyAmount()</code> <code>setTransactionCurrencyAmount()</code> <code>getClaimCurrencyAmount()</code> <p>For example:</p> <pre>_lineItem1.setTransactionCurrencyAmount(CurrencyAmount.UTIL.get(288, Currency.EUR))</pre> <p>Note that <code>getAmount()</code>, <code>setAmount(BigDecimal)</code>, and the <code>Amount</code> property are still available for backward compatibility and future single currency mode implementations.</p>

Issue ID	Description
CC-40186	Removed the “deprecatedDefaultExpression” attribute from the PaymentRecurrenceInput usage in NewPaymentInstructionsDV.pcf. All numeric attributes in PaymentRecurrenceInput now default to blank if not specified. The exception is month_rel_week_default which maps 1 to “first”, 2 to “second”, 3 to “third”, 4 to “fourth”, 5 to “fifth”, and anything else to “first”. Dates default to the current date if not specified or a blank value is specified.
CC-40350	Added option to submit files to SCM (source control management) system. With this change, the Subversion (SVN) version control module now requires that you enter the client root (Tools → Options → Version Control).
CC-40522	Modified the Subrogation Financials tab so that it interprets a null returned by a financial calculation as a zero (0). Previously, a null value would lead to non-display of certain amounts.
CC-40545	Added methods getTransactionsIterator() and getPaymentsIterator() to ReserveLineImpl.
CC-40587	Modified PaymentRecurrenceInputNode to expect string attributes with specification of the expected GScript expression value types: <ul style="list-style-type: none"> • recurringDefault - boolean • weeklyDefault - boolean • monthAbsDefault - boolean • inAdvanceDefault - boolean • weekCountDefault - int • monthAbsDayDefault - int • monthAbsCountDefault - int • monthRelWeekDefault - int • monthRelCountDefault - int • advanceDaysDefault - int • numChecksDefault - int
CC-40597	Modified InjuryIncident so that setting Received Treatment? to false clears the primary doctor and treatment type. Also, added an error message requesting that you set the doctor or treatment type if you set Received Treatment? to true and try to save without setting any values for these fields.
CC-40657	Modified the description of the Exposure Closed Validation Rules rule to read as follows: “This is the Exposure Closed Pre-Update Validation rule set. These rules are executed when an exposure is closed. They execute before data is committed and before the normal pre-update and validation rule sets. They enable you to raise validation errors or warnings if the exposure should not be closed.”
CC-40690	Added ability to view and edit typecodemapping.xml through Studio (in Other Resources).
CC-40851	Added ability to view read-only dm_* files through Studio (in Data Model Extensions → metadata).
CC-40924	Modified the Claim Search (Advanced Search) page to display non-searchable fields as unavailable when searching for archived claims.
CC-41049	Removed the hyphens in the tax ID when making an ISO Claim Search request.
CC-41092	Added ability to view and edit the following files through Studio (in Other Resources): <ul style="list-style-type: none"> • datamappingsplit.xml • datamappingtogether.xml • typelistmappingsplit.xml • typelistmappingtogether.xml
CC-41247	Added the following assignment methods: <ul style="list-style-type: none"> • assignToClaimOwner() • assignToClaimUserWithRole() • assignToPreviousOwner() • assignUserByProximityWithAssignmentSearchCriteria()
CC-41306	Augmented build.properties to include optional JVM memory parameters for use in running ClaimCenter on laptops with 2G of memory. See build.properties for example settings.
CC-41314	Added simple text editor to Studio for use with XML files editable in Studio, but which have no dedicated editor. This text editor is active if an external file editor is not specified.
CC-41512	Added ability (through a build.properties scm.regex statement) to define which files or file directories Studio should exclude in calculating the checksum for source-controlled files. (The build.properties file provides examples for the CVS and SVN source control systems.)

Issue ID	Description
CC-41585	Modified the ISO event messaging rules to safely handle exceptions involving request-scope variables. The ISO event messaging rules use a request-scope variable to indicate whether ClaimCenter should display a message to the user explaining that a message has just been sent to ISO. These rules check for exceptions when they access this variable, so that, for example, if the event messaging rules are fired from, say, a batch process, they do not simply crash. If the rules do receive an exception accessing the variable, they now treat the value as null (or just not set it). The variable only needs to be set if there is an actual user HTTP request, to trigger a message for the user.
CC-41586	Modified the ISO integration so that it looks for claim contacts on Incident as well as on Exposure. Formerly, integration logic only added contacts directly owned by the Exposure to the ISO message payload. With this change, the logic also adds any contacts owned by the exposure's incident as well.
CC-41748	Updated the ISO style sheet to the latest version (which includes an enhancement to mask the first five digits of social security numbers).
CC-41769	The <code>serverId</code> used to define server clusters in <code>config.xml</code> is now treated case insensitively.
CC-42103	The Server Tools page now contains a link to view the JProfiler.
CC-42331	Modified cache sizes for specific entity types in the base configuration to improve performance. The new cache information is: <ul style="list-style-type: none"> <code><param name="UserCacheSize" value="4000"/></code> <code><param name="GroupUserCacheSize" value="6000"/></code> <code><param name="ContactCacheSize" value="6000"/></code> <code><param name="TAccountCacheSize" value="4000"/></code> <code><param name="CheckCacheSize" value="4000"/></code> <code><param name="ClaimAccessCacheSize" value="10000"/></code> <code><param name="ClaimContactCacheSize" value="10000"/></code> <code><param name="ClaimContactRoleCacheSize" value="10000"/></code>

General Issues

The following are the primary issues corrected in this release:

Issue ID	Description
CC-18964	Fixed an issue in which the calendar screen did not update to reflect changes made to activities
CC-31467	Fixed an issue with client reflection on Set Reserves page to update Change amount and sum totals.
CC-33181	Fixed an issue in the Matter detail view that substituted the <code>HearingDate</code> for the <code>ArbitrationDate</code> value.
CC-35883	Fixed an issue where an error was generated when deleting a vehicle incident referenced by a Metro report.
CC-36172	Fixed an issue in which a query involving a large number of users in a group would generate an error. Added configuration parameter <code>MaxACLParameters</code> to specify when query optimization is invoked.
CC-39300	Fixed an issue in which a search in the AddressBook returned the number of results found (but not the data) even though the user did not have the requisite permissions to view the data.
CC-39340	Sample code for dynamic assignment is now visible and editable in Studio.
CC-39832	Fixed an issue where upgrading workers' comp related typelists would log an erroneous error.
CC-40197	Fixed an issue in which the round-robin assignment state was updated even if the assignment itself was not saved due to validation problems (for example, on an existing claim).
CC-40572	Fixed an issue where an upgrade did not fail if an extension for <code>BodyPart</code> was not present in the target installation.
CC-40645	Fixed an issue that allowed a retired exposure type to omit the <code>Category</code> element specifying that exposure type's incident type. You must now specify an incident type, even for retired exposure types.
CC-40755	Fixed an issue with <code>assignGroupByRoundRobin()</code> to correctly handle a null <code>GroupType</code> parameter.
CC-40783	Fixed an issue where <code>RecoveryImpl.isOffsetRecovery()</code> and <code>.getRecoveryBeingOffset()</code> do not work correctly when called from Event Fired rules.

Issue ID	Description
CC-40834, CC-40836, CC-40849, CC-40850, CC-40880,	Fixed the "Revert to Base" file management functionality used by the SCM (source control management) system. This functionality now works with files managed by the following Studio editors: <ul style="list-style-type: none"> • Rules editor • Entity Names editor • Web Services editor • Lines of Business editor
CC-40930	Fixed an issue that prevented moving or adding ClaimCenter widgets in the Studio PCF editor.
CC-41023	Fixed an issue in which using the Studio text search feature did not work if there was a period in the search string.
CC-41040	Fixed an issue with <code>PolicyLocations</code> with null Address that caused update problems. City and state are now required on the location's address.
CC-41179	Fixed a Studio issue in which typing in an editor activated the Studio code completion feature (dot completion), which, in turn, negated the keystrokes, requiring retyping of the characters.
CC-41388	Fixed an issue where deleting an authority limit profile would not delete the corresponding authority limits.
CC-41437	Fixed an issue that created a Null field mapper parsing error on update between ClaimCenter and ContactCenter.
CC-41564	Fixed an issue with an upgrade trigger that assumed there would be at least one exposure mapped to InjuryIncident. The fix, in part, moved <code>LocationAddress</code> from the base application model into <code>extensions.xml</code> . Important See "Potential to lose <code>LocationXXX</code> data during upgrade (CC-41555)" on page 334 for significant upgrade issues related to moving <code>LocationAddress</code> into <code>extensions.xml</code> .
CC-41574	Fixed errors with the <code>regen-pcf-mapping</code> tool.
CC-41781	Fixed an issue where the "-" link on the Claim History page would generate an error. The link is now no longer shown if it is not valid.
CC-41962	Fixed an issue where the Team page would display negative values.
CC-42019	Fixed an issue in which the Go to Type dialog (activated by Ctrl-N) did not work with <code>quickjump-config.xml</code> (the QuickJump resource).
CC-42072	Corrected an issue in which performing a "Verify All" or a "Verify Path" in Studio did not display the correct mode for errors/warnings in modal PCF files. Double-clicking on a verify error/warning incorrectly showed the default mode instead of the mode on which the error occurred.
CC-42609	Fixed an issue where retrieving an archived claim after upgrading would fail.
CC-42691	Fixed an issue where assigning a group by round robin failed when two processes were doing it against the same database.
CC-42795	Fixed an issue where a new resource context added to the <code>ResourceContext</code> typelist would not appear in the Resource Context dialog.
CC-42838	Fixed an issue where <code>EntityFactory</code> would throw an <code>IllegalStateException</code> .

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER/Software/Microsoft/Windows/CurrentVersion/Explorer/BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin configuration in ClaimCenter/config/config.xml.

Using a message handling plugin to deny a check containing a final payment causes an error (CC-27165)

Issue: Using a message handling plugin to deny a check containing one or more 'final' payments causes an error. Denying checks that contain only 'partial' and 'supplemental' payments via a message handling plugin works correctly.

Workaround: Use the IClaimFinancialsAPI to deny a check with a final payment instead of a message handling plugin. Guidewire may fix this issue in a future release.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Changing a coverage now fires the *Policy_Changed* event in addition to the *Coverage_Changed* event (CC-29564)

Issue: Changing a field on a Coverage now fires the `Policy_Changed` event, in addition to the `Coverage_Changed` event that has always fired.

Workaround: None.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in `filename`.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` method in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a Pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaimAPI depends on *SynchStateData*, which uses deprecated methods (CC-31611)

Issue: `IClaimAPI` depends on `SynchStateData`, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

An exception is thrown when trying to link a new contact to the address book from the new check wizard (CC-32732)

Issue: If you create a new contact in the New Check Wizard, and later attempt to link contacts to ContactCenter, you get an error.

Workaround: None.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the Checks screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To retrieve the payment's send date, both the Transactions and the Checks table would have to be searched; this would hamper performance and is not done.

Workaround: None.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns, then rerun the PCF Converter.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The Reset button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the Reset button an Initialize button.

After logout, a request to return to the login page creates a new session in Websphere (CC-36394)

Issue: On logout in WebSphere, the current session is destroyed but a new one is created. In ClaimCenter, logging out takes you back to the login page. Whenever a browser interacts with an application server, the server creates a session if none existed.

Workaround: On logout, go outside of the Guidewire application and enforce static content caching: the browser will reuse static objects in its cache (like the login page) and therefore minimize the unnecessary interactions with the application server. Alternately, have a web tier cache the static content. This is basically adding another caching layer (on top of the browser) and would reduce this phenomenon further.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you go to the ClaimCenter Desktop Activities page and repeatedly click Activities, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to Internet Explorer 7.

External Editors are not recognizing Tool / Option Settings in Studio (CC-40225)

Issue: External editors that are set in the Tools / Option menu are not being recognized by the Studio application when you click on `config.xml` or `profile.properties`. Also, opening some files with an external editor gives a

message asking whether to edit the file and place a copy of the file in the configuration module. A **No** answer still opens the file in the external editor.

Workaround: Copy the file to the configuration module and edit it there directly with the external editor.

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the Plan of Action > New Evaluation page will display an incorrect title (“New Evaluation” instead of “LOBName Evaluation”), and the page will not have a “LOBName Cost” field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use `ClaimEvaluationDetail` page instead of `ClaimEvaluationDetails<LOB>` page. Also, do not customize the `ClaimEvaluationDetail` page heavily.

Acrobat Sample document template does not allow extra fields (CC-40462)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

MaxBrowserHistoryItems gives an error (CC-40466)

Issue: In version 4.0.x, you could disable the Back button by changing the `MaxBrowserHistoryItems` parameter to 0. This is no longer possible in version 5.0.0.

Workaround: None.

FNOL wizard's Basic Info page has incorrect *post on change* behaviors (CC-40479)

Issue: On the **Basic Info** (second) page of the New Claim wizard, editing the **Reported By** contact and then switching to another contact can sometimes edit the wrong contact and can also commit multiple contact changes.

Workaround: None.

You cannot delete a workflow, and an invalid workflow prevents the server from starting (CC-40533)

Issue: It is not possible to delete a workflow. Further, if you invalidate a workflow in the process of editing it, the invalidated workflow will prevent the application server from (re)starting.

Workaround: Instead of deleting an obsolete workflow, make sure that your application never triggers it. Also, use Studio’s workflow editor to make sure all workflows are valid.

ClaimCenter uses an old version of Castor libraries (CC-40659)

Issue: Guidewire products still use an old version of the Castor library, and you cannot modify the version used. The version in ClaimCenter is bundled version 0.9.4.3. The current version of the Castor libraries is 1.2.

Workaround: You may use another version of this library within your plugin code. To use a different version of Castor, add its jar to the `modules/configuration/plugins/<plugindir>/lib` folder. All jar files located in the plugins directory are loaded with a different class loader than the one used by the ClaimCenter core code. This avoids any potential conflicts between versions of the libraries.

Potential to lose LocationXXX data during upgrade (CC-41555)

Issue: Guidewire changed the location of the LocationCity, LocationState, LocationStreet, and LocationZip fields between ClaimCenter 3.1.x, 4.0.x, 5.0.0, and 5.0.1:

- In ClaimCenter 3.1.x, these fields live on the exposure.
- In ClaimCenter 4.0.x, these fields live as extensions on MobilePropertyIncident.
- In ClaimCenter 5.0.0, Guidewire moved these fields into a separate LocationAddress object, in the core data model.
- In ClaimCenter 5.0.1, Guidewire changed LocationAddress into an extension on MobilePropertyIncident (in extensions.xml).

Upgrading from ClaimCenter 3.1.x or 4.0.x. Guidewire added a version check that should prevent data loss when upgrading to 5.0.1 from either 4.0.x or 3.1.x. It will look for the LocationXXX fields in incident or exposure (depending on whether you are upgrading from 4.0.x or 3.1.x). If any of those fields exist, then the version check will insist the following:

1. There is a LocationAddress field, of type Address, somewhere in the incident hierarchy
2. That any incidents/exposures with the LocationXXX fields set map to incident types that contain the LocationAddress field. That is, it checks that all LocationXXX fields that contain a value have a LocationAddress field to use in migrating the data.

If either of these checks fail, the version check will abort the upgrade and you must then follow the steps outlined in the workaround to solve the problem.

Upgrading from ClaimCenter 5.0.0. If you are upgrading from ClaimCenter 5.0.0, there is already a LocationAddress field. However, in 5.0.0, Guidewire placed the field in the core data model, while in 5.0.1 it is in extensions.xml. So, if your 5.0.1 extensions.xml does not contain a LocationAddress field, then the LocationAddress field could get dropped entirely during the upgrade process, possibly losing data. This is the case that the upgrade version trigger does not address.

Workaround: Depending on the release from which you are upgrading, you need to do the following:

- **ClaimCenter 3.1.x base release.** In 3.1.x, the LocationXXX fields were in the base data model, on exposure. As part of the upgrade, you will need to decide which exposure types map to which incident subtype. You must map any exposure types that have the LocationXXX fields set to an incident subtype that has the LocationAddress field. This ensures that the upgrade process has a place to move the data.
- **ClaimCenter 4.0.x base release.** In 4.0.x, the LocationXXX fields were in extensions.xml, in the incident hierarchy. You could choose where in the incident hierarchy to put these fields. In upgrading to any 5.0.x version, replace these fields in extensions.xml with LocationAddress, at the same level in the hierarchy.
- **ClaimCenter 5.0.0 base release.** In 5.0.0, the LocationAddress field was in the base data model, in MobilePropertyIncident. If you are upgrading to 5.0.x, you will need to make sure LocationAddress is in the extensions.xml file. Guidewire recommends that you put it in MobilePropertyIncident. However, if desired, it is possible—because it now lives in extensions.xml—to move it to a different place in the incident hierarchy.

“Verify All Resources” causes out-of-memory error (CC-40499)

Issue: In Studio, running the (Tools) Verify command with the Verify all Resources option selected generates a Studio out-of-memory error, if you do this a number of times without restarting Studio.

Workaround: Guidewire has implemented an “incremental” verification strategy that verifies any changes made since the last verification. With this release, you can now make changes in Studio or retrieve the latest files from source control, for example, and then do an incremental verify to see if these changes have broken anything. Guidewire recommends that you do not use the Verify All Resource command on a regular basis, but, instead, use the Verify command in its place. Note that there are some changes that incremental verify cannot detect, such as changes to external Java files. If these sorts of changes occur, a Verify All Resources command is necessary.

Reserve amounts do not refresh properly when increasing reserves while creating a new check (CC-31371)

Issue: If you are creating a check and need to increase the reserves in another session to make the payment, ClaimCenter does not refresh the reserve amounts when you increase the reserves unless you close the new check creation and open it again.

Workaround: You must go back one step and forward again (in the check wizard) to force a data bundle refresh and to update the reserves amount in step 2 of the unsaved check wizard.

Messaging plugin not initialized unless it explicitly implement InitializablePlugin (CC-42481)

Issue: The Messaging plugin does not initialize properly unless it explicitly implements the `InitializablePlugin` interface.

Workaround: All Messaging plugins must implement `InitializablePlugin`.

Purge of “inactive” messages recommended before upgrade (CC-42573)

It is important to purge completed (inactive) messages before upgrading to a new version. Doing so reduces the complexity of the upgrade. You can purge completed messages using either the command line tool command:

```
messaging_tools -purge date
```

Replace “date” with a date, and this tool deletes completed messages that are older than that date. You should periodically use this command to purge old messages to avoid the database from growing unnecessarily. Alternatively, you can use the web service API `IMessageToolsAPI.purgeCompletedMessages(Calendar date)`.

For upgrades, always perform this message purge on the old server **before** upgrade begins, so that the upgrader does not attempt to convert those database table rows.

Attempting to add Lost Time/Work Status on claim with no EmploymentData results in error (CC-42652)

Issue: If you attempt to add a Loss Time/Work Status entry on the **Loss Details** page—without an associated `EmploymentData` entity—ClaimCenter generates the following when you click the **Add** button:

```
EvaluationException: "EmploymentData" evaluates to null in the expression
"EmploymentData.addToWorkStatusChanges( WorkStatus )", so the function "addToWorkStatusChanges"
cannot be evaluatedStack Trace:com.guidewire.commons.gscript.parser.EvaluationException:
"EmploymentData" evaluates to null in the expression "EmploymentData.addToWorkStatusChanges(
WorkStatus )", so the function "addToWorkStatusChanges" cannot be evaluated
...
```

Workaround: You must first add `EmploymentData` to the claim before adding Lost Time/Work Status details.

Verifying ClaimPrintout.pcf gives Null Point Exception (CC-42586)

Issue: Attempting to verify `ClaimPrintout.pcf`, or the folder in which it is located, gives an erroneous verification warning. You can safely ignore this exception. You can click on `ClaimPrintout` in the **Resources** tree and select **Verify View** to ensure that `ClaimPrintout.pcf` does not contain an error.

Workaround: Guidewire is aware of the issue and plans to address it in a future release.

Upgrading Contact Center from ClaimCenter 3.0.x to ClaimCenter 5.0.x or later requires rules merging before trying to create a new contact (CC-42699)

Issue: After upgrading ContactCenter from ContactCenter 4.0.x to 5.0.0 or higher, certain rules need to be merged into the upgraded environment before you try to create any contact in that upgraded ContactCenter environment. If you do not do this, ContactCenter will fail create the contact. For example, you need to merge rule

`V.ABContactValidationRules.DefaultValidationRules.RequirePastDOB` where `Libraries.Date` in the Rules Action needs to be changed to `gw.api.util.DateUtil`.

Workaround: Search for rules that use any of the old-style Libraries functions and modify them to use the new GScript functions *before* attempting to create any contact in the upgraded ContactCenter environment.

Transaction `runWithNewBundle` Does Not Follow Links (CC-42654)

The GScript API that runs a block of code with new bundle (`gw.transaction.Transaction.runWithNewBundle(...)`) has some limitations. Even if you call `newbundle.add(yourentity)`, the add method does not follow the graph links to other entities to which it links. You must manually add each sub-object to the new bundle, using the call `bundle.add(linkedEntity)` for each entity. This includes all foreign key references to other entities as defined in the built-in data model files, but also includes customer data model extensions that include foreign key references.

You can work around this by committing all entities in the current bundle instead of using this API. This may be an issue when designing implementing a custom web service. Committing all entities in the current bundle includes all entities serialized into the bundle as part of the web service request because they were parameters to a web service method. There may be cases where entities that are web service API parameters you do **not** want commit to the database.

One workaround is to redesign the web service API so instead of taking entities (the entities you do not want to commit) it takes GScript class instances instead of Guidewire entities. Your custom GScript classes can provide only the fields that you want to pass in your integration code, which may be a much smaller subset of fields in some cases. Because they are GScript classes and not entities, committing the entire bundle with `gw.transaction.Transaction.getCurrent().commit()` will skip that data that you choose not to commit to the database.

Creating a new document from a template results in a `ClassNotFoundException` (CC-42887)

Issue: If you implement the `IDocumentProduction` plugin in Java, then when you create a new document from a template you may encounter a `ClassNotFoundException`.

Workaround: Copy the file `ClaimCenter/modules/cc/config/plugin/ExternalMappings.properties` to `ClaimCenter/modules/configuration/config/plugin`. Then add the following line to the copied file:

```
gw.api.document.DocumentCreationInfo = gw.api.document.DocumentCreationInfo
```

In addition, copy the file `ClaimCenter/modules/cc/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf` to `ClaimCenter/modules/configuration/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf`. In the copied file, replace the following line:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    DocumentProduction.createDocumentsSynchronously(DocumentCreationInfo.DocumentTemplateDescriptor.
      TemplateId, gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

with the following:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    util.document.DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.
      DocumentTemplateDescriptor.TemplateId,
      gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```


Documentation-Related Issues

Manually merging WC- and MP-prefixed typelists (CC-42663)

Issue: The section in the *ClaimCenter 5.0.1 Upgrade Guide* called “Manually merge WC- and MP-prefixed Typelists” contains some outdated information. First, there are no MP-prefixed typelists. Second, most of this merging is now done for you by a configuration upgrade step as of 5.0.1. However, your intervention may be required if these WC-prefixed typelists were modified such that the same typecode in the two lists have conflicting Name attributes.

Workaround: After running the ClaimCenter 5.0.1 configuration upgrade, and before proceeding to the manual merge phase, you should examine the five typelists (those without the WC prefix) and be sure there are no Warning comments which indicate a conflict in Name. If present, the warning will look similar to the following:

```
<!-- Warning typecode 'xxx' is present in both WCExampleTypelist and ExampleTypelist  
but have different names -->
```

Note: This warning will appear in ExampleTypelist, not WCExampleTypelist. The WCExampleTypelist will remain unchanged (in case it is used in extension fields).

You either need to overwrite the Name of the 'xxx' typecode in ExampleTypelist with the Name of the 'xxx' typecode from WCExampleTypelist. Or if the 'xxx' typecode is used to mean different things in the two typelists, you need to fix your data to use different typecodes in this case, so the distinction is not lost when the typelists are merged. (For help in this rare case, please consult with Guidewire Support.)

ABAddress Renamed to Address

ContactCenter 4.0.x contained the entity, ABAddress. In ContactCenter 5.0.0 and newer releases, Guidewire renamed this entity to simply Address. Existing ClaimCenter and ContactCenter documentation not only does not make this clear, but incorrectly references the old name for the entity, ABAddress.

For a ContactCenter upgrade, Guidewire does not automatically update any extensions.xml ABAddress extensions to Address. You must manually update extensions.xml to change the extensions to be on the Address entity instead of ABAddress.

As part of the name change to Address, Guidewire implemented two other changes:

1. Guidewire removed the Geocodeable interface. As it was not an entity, it was not extensible. However, its removal could create a potential problem if custom rules or GScript code made references to it. With this change:
 - APIs that formerly returned a Geocodeable object/entity now return an actual Address entity
 - APIs that formerly required a Geocodeable object/entity as a parameter now require an actual Address entity
2. Guidewire also removed the NonPersistentAddress entity. APIs that referenced this entity now behave in a similar fashion to those that referenced a Geocodeable object/entity.

With these changes:

- There is no longer any way to support extensions only for non-persistent addresses.
- There is also no longer any way to support extensions only to addresses for local user contacts in ContactCenter.

Upgrading Libraries and the Java Method invokeLibraryMethod

In an earlier release of ClaimCenter, GScript library support was removed and replaced with code implemented as GScript classes (for general utilities) and GScript enhancements (for extensions methods to be added to enti-

ties). As a consequence of this change, Java code implementing Java plugins or other Java called from GScript could not call Java libraries or access static methods and static fields on Java classes.

ClaimCenter 5.0.1 adds additional features that make it easier to call GScript utilities from Java. This is especially relevant for customers upgrading from earlier versions of ClaimCenter that used the `invokeLibraryMethod` from Java to call *libraries*, which no longer exist in exactly that form.

As of ClaimCenter 5.0.1, GScript classes and instances of GScript classes are now exported to Java through the entity libraries. For example, a GScript class called `MyUtils` will be exported into Java into the package `external.*` as a class called `MyUtils`. All methods and properties are dynamically translated from GScript to Java, and back again as needed.

You can use these improvements to the entity libraries in ClaimCenter 5.0.1 to replace former usages of `invokeLibraryMethod` with direct calls to your GScript classes that now encapsulate former libraries.

Refer to earlier in these releases notes for more information about entity library changes.

Guidewire ClaimCenter 5.0.2 Release Notes

Release 5.0.2.12

Copyright © 2008 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire ContactCenter, Guidewire Insurance Suite, and the Guidewire logo are trademarks 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 documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 5.0.2.12.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact Guidewire Customer Support. You may contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For installation information, please refer to the *ClaimCenter Installation Guide*.

Note Regarding Rules After Upgrade (CC-44608)

When upgrading from a previous ClaimCenter 5.0.x release, you will see the base 5.0.2 rules appear in Studio, rather than the preserved base unmodified rules from your previous release. This is a change in upgrade behavior from previous releases, but since these are base uncustomized rules, there should be little to no impact on your upgrade or your system behavior.

Note: This change does not impact your customized rules. All customized and modified rules from your previous release will be properly upgraded to your destination release.

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Improvements
- General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 5.0.0 and ClaimCenter 5.0.1, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- `gw.api.*` GScript API
- typelists
- display keys

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.1 to 5.0.2

- 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/platform` directory, [click here](#).

Base Resource Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.1 to 5.0.2

- To view a report of the changes in the base resources in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base resources in the `modules/core` directory, [click here](#).
- To view a report of the changes in the base resources in the `modules/platform` directory, [click here](#).

Improvements

The following are other improvements made to ClaimCenter for this release:

Issue ID	Description
Database Issues	
CC-41795	The QuickStart database (H2) now supports auto-upgrading (you do not need to drop and recreate the database) for the following changes: <ul style="list-style-type: none"> • adding a column to an existing entity • adding an entity • adding a typelist • adding a typecode to an existing typelist
Financials Issues	
CC-42889	New payments now allow <code>setAsEroding()</code> and <code>setAsNonEroding()</code> , even if they are Submitting.
General Issues	
CC-42562	In GScript, type filters are now available as <code>TF_*</code> static properties on typelist types.
CC-42814	Text in the Alert Bar now wraps if necessary.
Internationalisation Issues	
CC-41741	Added the "imeMode" attribute to all PCF inputs and cells. This controls the state of an Input Method Editor (IME) for entering Japanese language characters. Setting this to "active" turns on IME on focus; "inactive" turns off IME on focus; and not setting this at all keeps the IME setting the same.
PCF Editor Issues	
CC-43700	You can now cut, copy, and paste Variables, Requires, ReflectConditions, and more in the Page Configuration editor.
Studio Issues	
CC-34976	Studio now allows you to configure a diff application for diffing files with their base versions.
CC-38048	In Studio, <code>@Throws</code> exceptions are now included in WebService endpoint documentation.
CC-42225	Fixed an issue where opening the Go To Type dialog would take a long time after reverting a resource to base.
CC-43696	If you Ctrl+Click or Ctrl+B to move down to a method in a file, and then go back, you now go back to the previous point in the same file instead of the previous file.
CC-43767	When an object is open in the Studio debug window, you can now type a letter to jump to the first sub-object in the list that begins with that letter.
CC-43870	Implemented various improvements to the debugger UI in Studio.
Web Issues	
CC-42146	Having a <code>PostOnChange</code> attribute set to false and an <code>OnChange</code> attribute with a value is now a warning rather than an error.
Workflow Issues	

Issue ID	Description
CC-42359	The following workflow-related methods are now overloaded to accept a workflow version number: <ul style="list-style-type: none"> start() startAsynchronously() createSubFlow(WorkflowIntSubtype)

General Issues

The following are the primary issues corrected in this release:

Issue ID	Description
ContactCenter Integration Issues	
CC-42257	The ABAAuthenticationPlugin is now disabled by default.
Administration Issues	
CC-41204	Fixed an error with running the Rule Execution Report on the Internal Tools page.
CC-42438	The server name (server attribute) in the work-queue configuration in config.xml is now treated case insensitively.
Assignment Issues	
CC-43557	Added method owner.getOrCreateUserRoleAssignmentByRole().
Claim Issues	
CC-40188	Fixed an issue where WC Benefits calculations were not triggered for all new WC claims.
CC-42671	Fixed an issue where the SIU trigger "days between loss date and notice date" did not work.
CC-43033	Fixed an issue that would occur when you attempted to update a medical case management with a null MedicalContactStatus.NextFollowUpDate value.
CC-43525	Fixed an issue where the 'Actual % Recovered' value on the Subrogation page was not correct.
CC-44073	Fixed an error where switching the policy during the New Claim wizard would cause an exception. Also deleted the following unused PCF files: <ul style="list-style-type: none"> web/pcf/claim/FNOL/QuickClaimDV.Auto.pcf web/pcf/claim/FNOL/QuickClaimDV.Pr.pcf web/pcf/claim/FNOL/QuickClaimDV.WC.pcf
Configuration Issues	
CC-43001	The ContactContact entity is now extendable.
Contact Issues	
CC-43112	You can no longer remove an injured party if there is an injury incident and it is related to an exposure.
Claim Wizard Issues	
CC-42652	Fixed an error that could occur when you attempted to add a Loss Time / Work Status entry on the Loss Details page.
Database Upgrade Issues	
CC-43623	Fixed an issue where during an upgrade a typelist with case change in its name would result in an incorrect error reported.
Desktop Issues	
CC-43883	Fixed an error that would occur when viewing the calendar with a month containing only four weeks.
Document Management Issues	
CC-42474	Fixed an error that would occur when canceling the creation of a document using a template for an activity.
Financials Issues	
CC-38867	Fixed an issue on the Claim Search screen where the Search For Date and Financial Value fields would show only the value that was previously chosen and <none selected> as options.
CC-40021	Fixed an error that was caused by transferring a check that was the only member of a check group.

Issue ID	Description
CC-40039	Added the following consistency/integrity checks: <ul style="list-style-type: none"> Transaction.TransToClaimExchangeRate is not null for Transactions with a non-default currency staging table ExchangeRates are not shared between Transactions
CC-41461	Fixed an issue where a foreign-exchange adjustment was made to an escalated payment and was transferred, but the transfer amount (claim amount) of the payment was not also updated.
CC-41471	Fixed an issue where a transferred check's claim amount is altered in the transferred claim with a prior foreign-exchange adjustment made to the check.
CC-42398	Corrected the following behavior with financials-related methods: <ul style="list-style-type: none"> setPaymentType() now changes NonEroding to false if the payment type is not Supplemental. setNonEroding() now attempts to set its associated check's payment to eroding or non-eroding as long as the check exists.
CC-42543	Fixed an issue where editing final payments would result in an incorrect error message about exceeding available reserves.
CC-42587	Fixed an error that would occur when calling Check.getPrevPaidOrScheduledServicePeriod() when ServicePeriodStart was null.
CC-42635	Fixed an error that would occur when you would create a reserve with multicurrency and no exchange rates were available.
CC-42676	Fixed an issue where a primary check's CheckRpt.GrossClaimAmount for a multicurrency and multipayee check was not updated correctly when the exchange rate or Currency was changed.
CC-42681	Fixed an issue where closing an exposure would cause inconsistent escalation of checks.
CC-43047	Fixed an issue that would occur when you create a reserve, get a validation error, and then click Add anyway.
CC-43274	Fixed an issue where the Pay To field was not being updated properly in the Quick Check wizard.
CC-43438	Fixed an issue where recoding a payment would log a message that the payment was deleted rather than recoded.
CC-43448	Fixed an issue where deleting a check would result in a history entry with an invalid link.
CC-43521	When a check is transferred or a payment is recoded, it is now possible via changes to PCF files to configure the PaymentType and ErodesReserves properties of the new onset payments. These values were previously overwritten in Java while being saved, and so not configurable.
CC-43577	When a payment is denied, instead of denying the offset reserve, ClaimCenter now creates an offset to the offset reserve, and disassociates the first offset reserve from the denied payment.
CC-43589	Fixed an issue where check-related history descriptions did not contain the correct check amount.
CC-43610	Fixed an issue where a supplemental payment that was recoded/transferred to an open exposure would be set to non-eroding instead of eroding.
CC-43697	Fixed an error that would occur when creating a bulk invoice with a negative amount item.
CC-43714	Fixed an issue where a bulk invoice's check number would not be associated with multiple placeholder checks.
CC-43718	Fixed an issue where in the New Check wizard changing the payment from Supplement to Partial would not change the Non-Eroding property back to Eroding.
General Issues	
CC-39560	Fixed an issue where multiple consecutive spaces in text fields were being saved as non-breaking spaces.
CC-40073	Fixed an issue where malformed XML was not being reported with PCF verification errors.
CC-41360	The method String.isEmpty() has been removed. Instead, use the String.HasContent property. For example, instead of: <pre>if(not String.isEmpty(str)) { ... }</pre> use: <pre>if(str.HasContent) { ... }</pre>
CC-42513	Fixed an issue with GScript code that calls a Java method that returns an array whose component type is not an external class or core Java class.

Issue ID	Description
CC-42601	<p>The methods <code>getDisplayName(ILocale)</code> and <code>getDescription(ILocale)</code> are now available on <code>TypeKeys</code>, both in GScript and Java.</p> <p><code>ILocale</code> is a new type that has a GScript form and a Java form.</p> <p>In GScript:</p> <pre>var enUSLocale = gw.i18n.ILocale.EN_US</pre> <p>In Java:</p> <pre>com.guidewire.external.i18n.ILocale enUSLocale = com.guidewire.external.i18n.Locale.EN_US;</pre> <p>Note that in Java, there are two classes:</p> <ul style="list-style-type: none"> <code>com.guidewire.external.i18n.ILocale</code> is the interface, instances of which are required by the locale-aware methods on <code>com.guidewire.external.typeList.TypeKey</code>. <code>com.guidewire.external.i18n.Locale</code> is the enum that implements the interface. Its constants are generated based on the your configured locales.
CC-42621	Fixed an error that would occur in GScript when a static inner class referenced a static method on its container.
CC-42690	<p>Removed the following invalid and disabled plugin configuration files:</p> <ul style="list-style-type: none"> <code>cc/config/plugin/registry/DBAuthenticationPlugin.xml</code> <code>cc/config/plugin/registry/IBaseURLBuilder.xml</code> <code>cc/config/plugin/registry/IMessagingNotification.xml</code>
CC-42859	Java classes in GScript, located in the <code>plugins/GScript/classes/</code> or <code>plugins/GScript/lib/</code> folders, can now be accessed on Linux.
CC-42869	Fixed an error that would occur when drilling down from a row tree on anything except the first level.
CC-43015	Fixed an error that would occur when using the <code>IDocumentProduction</code> Java plugin.
CC-43164	Fixed an issue in Studio where opening a file that contained an extraneous linefeed at the bottom would result in an autosave file to be generated, which caused the checksum to fail upon the next startup.
CC-43188	Fixed an error that would occur when <code>regen-toolkit</code> was run for external entities with inner classes. Inner classes are now ignored.
CC-43227	For the <code>TypeKey</code> class, the <code>Retired</code> and <code>Priority</code> properties used to return <i>box</i> types (that is, <code>java.lang.Boolean</code> and <code>java.lang.Integer</code> , respectively), but now they return the primitive types <code>boolean</code> and <code>int</code> . GScript generally handles <i>box</i> -to-primitive conversions seamlessly, but customers may run into compile errors with this change. In particular, you won't be able to call methods on the values returned by these properties, since they're now primitives, rather than objects. This is probably rare, as the methods that you would call on the <i>box</i> objects should be easily convertible into operators.
CC-43460	Fixed an issue where a commented-out PCF element would still be checked for proper syntax (and generate an error if it was invalid).
CC-43536	Fixed an issue where admin data export would run out of memory.
CC-43692	Fixed an issue where double-clicking to select text ignored prepended underscores in variable names.
CC-43719	Fixed several issues that would cause an empty dialog box to appear during <code>Verify</code> .
CC-43773	Fixed an issue where the auto-complete suggestion popup incorrectly showed the type of an array field.
CC-44118	Fixed an issue where <code>FinancialsSummaryCells</code> could not be reordered within a <code>FinancialsSummaryRow</code> .
CC-44119	Added the following attributes to <code><DualAxisDataSeries></code> : <code>lowerBound</code> , <code>lowerMargin</code> , <code>tickUnit</code> , <code>upperBound</code> , <code>upperMargin</code> , <code>useWholeNumbers</code>
CC-44397	Fixed an issue in GScript with incorrect coercions of floats and big decimals.
CC-44673	Fixed an issue that would occasionally cause an exception during evaluation of an if-then statement.
Internationalisation Issues	
CC-41243	The localization export result file is now sorted case insensitively.
CC-44212	Fixed an error where <code>regen-toolkit</code> would fail when Japanese Characters were used as typecodes.
Integration Issues	
CC-42855	Fixed a runtime WSDL parsing error that would occur upon deserialization of a SOAP response message.
CC-43071	Fixed an issue where the <code>Skip First</code> button on the <code>MessageControlForDestinationList</code> page did not work when there was only one message in error, retry, or inflight status.
CC-43588	Fixed an issue where custom events were being suppressed when added in <code>PluginCallbackHandler</code> .

Issue ID	Description
Metro Issues	
CC-42708	The Metro FireAutoReport now has a request type of X.
CC-42736	Fixed an issue where a document would not be shown when a "No Report" response was received from Metro.
CC-42782	Removed the phone number from the Driver1 element in a Metro request; this is not part of Metro's schema.
PCF Editor Issues	
CC-42291	The Title Bar element has moved to the Panel Layout group in the Page Configuration editor.
CC-43694	Fixed an issue where the Tab key would not respond correctly when creating a variable or required variable.
CC-43698	Fixed several issues with copy and paste working properly in the Page Configuration editor.
CC-43705	Fixed an issue where switching between tabs would show the Properties window for the wrong tab.
PolicyCenter Integration Issues	
CC-44305	Fixed a PluginException generated when trying to retrieve a newly created WC policy from PolicyCenter.
Policy Issues	
CC-41930	Fixed an error that would occur when editing a policy and selecting no value for both Additional Insured and Excluded Parties.
Printing Issues	
CC-43063	Fixed an issue where printing a List View with multiple Row elements would result in an exception. Only rows that are part of the row iterator will be printed.
Reporting Issues	
CC-43832	Restored the ability to allow Report Explorer to perform additional sorting and filtering.
CC-43878	Removed blank lines in the Oracle scripts for reporting.
SOAP API Issues	
CC-39753	Studio now supports web services for JAX-WS (Java Glassfish) WSDLs.
CC-41654	Fixed an error that would occur when calling the method IDataExtractionAPI.findClaimAndRenderWithTemplate(ByName).
CC-44132	Removed the 'hashCode' and 'equals' methods from SOAP GScript client types (for example, soap.Map-Point.entity.Foo no longer overrides the default 'hashCode' method). This more closely matches the behavior of the SOAP client in your toolkit (gw-soap-xx.jar) and the behavior that AXIS has when a WSDL2Java task is used to create a Java client for a web service.
Server Issues	
CC-42592	Fixed an issue where reloading PCF files would not always reload modified files.
CC-43669	Fixed an issue where the QuickStart server would not run on ports between 8800 and 8900.
Studio Issues	
CC-36479	Fixed an issue where copying a PCF element containing required variables would preserve the required variables in the pasted destination, even if they were not allowed.
CC-40499	Fixed an issue where verifying all resources multiple times would cause an out of memory error.
CC-40971	The User/Group selector in the Script Parameters dialog is now disabled when not connected to a server.
CC-41794	Fixed an error that would occur when moving or copying rule trees between rule sets.
CC-42065	Typelists and typekeys in Studio are now limited to 22 characters in length.
CC-42230	Fixed an issue where creating a new class would max out CPU resources.
CC-42267	Fixed an issue where the default Studio XML editor would mark a file as edited when you declined to edit it.
CC-42297	Fixed the behavior in the Go To Type dialog in Studio.
CC-42449	Fixed an issue where verifying GScript would apply to a previous version saved on the disk rather than the version visible in the editor. Verify now causes an implicit save.
CC-42452	Fixed an issue where a GScript function that references pcf.api would cause Studio to generate an error and exit upon startup.
CC-42505	Fixed an issue with incorrect code highlighting when editing a PCF element.
CC-42506	Fixed an issue where drop-down buttons would not always display correctly in Studio.

Issue ID	Description
CC-42516	Fixed an issue where moving a resource in Studio would not refresh the resource tree.
CC-42586	Fixed an error that would occur when verifying all resources in Studio.
CC-42697	Fixed an issue where Studio would sometimes not open a shared section after you double-clicked on it.
CC-42718	Studio now enforces that PCF file names must be unique within all PCF file names.
CC-42758	Fixed an issue where GScript enhancements could not be found using Ctrl+N search.
CC-43117	Fixed an issue where Studio prevented you from pasting invalid text into a name text box (such as for a new package). Now it replaces invalid characters with underscores.
CC-43553	Fixed an issue where if the SCM executable failed, there was still an empty file created in modules/configuration/resources/classes.
CC-43625	Fixed an issued where certain GScript classes could not be edited.
CC-43681	Fixed several auto-complete issues.
CC-43686	Fixed an issue where the parameter description dialog would not appear when there were later errors in a function.
CC-43687	Fixed an issue where typecode completion help would not show up after == and a space.
CC-43688	Fixed an issue where autocompletion for a typekey autocomplete would not show up when an invalid string value was used on the right hand side of an == expression.
CC-43689	Fixed several issues with value completion.
CC-43699	Fixed an issue where auto-completion on an unqualified local method would place the cursor after the parentheses.
CC-43707	Fixed an issue where typing Ctrl+N would not list PCF elements with the default mode being a multi-mode element.
CC-43728	Fixed an error that would occur when setting an external editor and omitting the file extension.
CC-43730	Fixed an issue where Studio added an extraneous brace character when you pressed Enter.
CC-43743	The PCF folder no longer collapses after you delete a PCF node.
CC-43752	Fixed an error that would occur when a menu item was dragged and dropped onto itself.
CC-43757	You can now drag a widget from a menu item popup onto a different menu container.
CC-43759	Fixed an error that would occur when trying to auto-fix (Alt+Enter) a case warning in code block.
CC-43769	Improved the reliability of Ctrl+Space code completion.
CC-43770	Fixed an error that would occur when adding a filter to a CoverageType.
CC-43774	In Studio, Ctrl+Right and Ctrl+Left now work even when a completion popup is visible.
CC-43817	Improved speed of auto-completion in Studio.
CC-43821	Fixed an error that would occur in a header cell when the iterator's entry name hadn't been specified yet.
CC-43829	Fixed GScript errors that were shown when a variable had the same name as an entity.
CC-43853	Fixed an error that would occur when deleting a PanelRef from a popup.
CC-43893	Leaving a PCF view and then returning now returns you to the point where you left.
CC-43894	Fixed an issue where double-clicking a verify or search result to go to a PCF widget would work only the first time.
CC-43933	Fixed an error where you would be prompted multiple times to open the rule to edit.
CC-43935	In Studio, Copy is now undoable.
CC-43944	Fixed an issue where Studio would display the Paste on the context menu when the clipboard did not contain data in the proper format.
CC-43952	Fixed an error in Studio that would occur when attempting to import a non-existent file.
CC-43963	Fixed an issue where a smart fix popup in Studio would prevent clicking in a Javadoc popup under it.
CC-43966	Fixed an issue where renaming a rule would silently create a new rule in the base module, which caused checksum failure and server failure to start.
CC-44064	Fixed an issue in Studio where Ctrl+Space code completion for a type would delete the method after the type.
CC-44092	Fixed an issue where a Studio panel would show a yellow warning where there was no problem with the code.

Issue ID	Description
CC-44104	Fixed an issue where Studio would complain that the return type of <code>CurrentLocation.getCurrentStepId()</code> is null when it should return String.
CC-44107	Automatically turning a text string into a display key in Studio now also results in the string being replaced inline.
Toolkit Issues	
CC-42193	Fixed an error that would occur when running regen-toolkit in a path that contained a space.
Validation Issues	
CC-42825	Fixed an error where warning messages were displayed in the Validation Result page after errors were fixed.
Web Issues	
CC-42600	Fixed an issue where keyboard shortcuts would not work for a button that only opens a submenu.
CC-42879	Fixed an issue where a commented-out PCF element would still be checked for proper syntax (and generate an error if it was invalid).
CC-43185	Removed the duplicate file <code>cc/config/web/pcf/claim/newclaimwizard/DuplicateClaimSearchResultsLV.pcf</code> .
CC-43357	Fixed an issue where specifying a non-existent class for the <code>contentType</code> of a RowTree element would not generate an error.
CC-43621	Removed keyboard shortcuts that used the key combination <code>Alt+Shift+D</code> . These were incompatible with Internet Explorer 7.
Workflow Issues	
CC-41501	The <code>WorkflowType</code> typelist is no longer used, and has been removed.
CC-41719	Fixed an error where the Resume option on the Manage Workflows page would not work when the workflow was in Error state.
CC-41720	On the Manage Workflows page, the Suspend option is no longer shown when the workflow is in Error state.
CC-41928	A warning is now generated if you add <code><events/></code> to a subtype of an internal entity that doesn't generate events.
CC-44190	The Workflow Statistics page is now available.

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the `[HKEY_CURRENT_USER/Software/Microsoft/Windows/`

CurrentVersion/Explorer/BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin configuration in ClaimCenter/config/config.xml.

Using a message handling plugin to deny a check containing a final payment causes an error (CC-27165)

Issue: Using a message handling plugin to deny a check containing one or more 'final' payments causes an error. Denying checks that contain only 'partial' and 'supplemental' payments via a message handling plugin works correctly.

Workaround: Use the IClaimFinancialsAPI to deny a check with a final payment instead of a message handling plugin. Guidewire may fix this issue in a future release.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in scheduler-config.xml for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the progressInterval attribute in the <workqueue> element.

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Changing a coverage now fires the *Policy_Changed* event in addition to the *Coverage_Changed* event (CC-29564)

Issue: Changing a field on a Coverage now fires the *Policy_Changed* event, in addition to the *Coverage_Changed* event that has always fired.

Workaround: None.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the *ab_abaddress* table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` method in your rules, and the *ExposureType* that you pass only maps to a single *LossPartyType*, then that *LossPartyType* will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no *PrimaryCoverage* or *LossParty*.

Workaround: You should add a Pre-update rule that sets the *PrimaryCoverage* and *LossParty* fields to a non-null value.

IClaimAPI depends on *SynchStateData*, which uses deprecated methods (CC-31611)

Issue: *IClaimAPI* depends on *SynchStateData*, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

An exception is thrown when trying to link a new contact to the address book from the new check wizard (CC-32732)

Issue: If you create a new contact in the New Check Wizard, and later attempt to link contacts to ContactCenter, you get an error.

Workaround: None.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the **Checks** screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To retrieve the payment's send date, both the Transactions and the Checks table would have to be searched; this would hamper performance and is not done.

Workaround: None.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns, then rerun the PCF Converter.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The Reset button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the Reset button an Initialize button.

After logout, a request to return to the login page creates a new session in Websphere (CC-36394)

Issue: On logout in WebSphere, the current session is destroyed but a new one is created. In ClaimCenter, logging out takes you back to the login page. Whenever a browser interacts with an application server, the server creates a session if none existed.

Workaround: On logout, go outside of the Guidewire application and enforce static content caching: the browser will reuse static objects in its cache (like the login page) and therefore minimize the unnecessary interactions with the application server. Alternately, have a web tier cache the static content. This is basically adding another caching layer (on top of the browser) and would reduce this phenomenon further.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you go to the ClaimCenter Desktop Activities page and repeatedly click Activities, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to Internet Explorer 7.

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the Plan of Action > New Evaluation page will display an incorrect title ("New Evaluation" instead of "LOBName Evaluation"), and the page will not have a "LOBName Cost" field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use ClaimEvaluationDetail page instead of ClaimEvaluationDetails<LOB> page. Also, do not customize the ClaimEvaluationDetail page heavily.

Acrobat Sample document template does not allow extra fields (CC-40462)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in SampleAcrobat.pdf.descriptor and SampleAcrobat.pdf files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

FNOL wizard's Basic Info page has incorrect *post on change* behaviors (CC-40479)

Issue: On the **Basic Info** (second) page of the New Claim wizard, editing the **Reported By** contact and then switching to another contact can sometimes edit the wrong contact and can also commit multiple contact changes.

Workaround: None.

You cannot delete a workflow, and an invalid workflow prevents the server from starting (CC-40533)

Issue: It is not possible to delete a workflow. Further, if you invalidate a workflow in the process of editing it, the invalidated workflow will prevent the application server from (re)starting.

Workaround: Instead of deleting an obsolete workflow, make sure that your application never triggers it. Also, use Studio's workflow editor to make sure all workflows are valid.

After upgrading from ClaimCenter 3.1, the doctor is no longer shown in medical detail exposures (CC-40595)

Issue: ClaimCenter 3.1 displayed the value `Exposure.Doctor`, but later versions of ClaimCenter display the value `Claim.FirstIntakeDoctor`. During upgrade, the data is not migrated from the old field to the new field, resulting in the upgraded ClaimCenter version showing a blank instead of the real value.

Workaround: Add another PCF field in ClaimCenter 5.0 on the **Medical Details** page below the current **Doctor** field for `Exposure.Claim.FirstIntakeDoctor`. This will display the upgraded data.

ClaimCenter uses an old version of Castor libraries (CC-40659)

Issue: Guidewire products still use an old version of the Castor library, and you cannot modify the version used. The version in ClaimCenter is bundled version 0.9.4.3. The current version of the Castor libraries is 1.2.

Workaround: You may use another version of this library within your plugin code. To use a different version of Castor, add its jar to the `modules/configuration/plugins/<plugindir>/lib` folder. All jar files located in the plugins directory are loaded with a different class loader than the one used by the ClaimCenter core code. This avoids any potential conflicts between versions of the libraries.

Potential to lose LocationXXX data during upgrade (CC-41555)

Issue: Guidewire changed the location of the `LocationCity`, `LocationState`, `LocationStreet`, and `LocationZip` fields between ClaimCenter 3.1.x, 4.0.x, 5.0.0, and 5.0.1:

- In ClaimCenter 3.1.x, these fields live on the exposure.
- In ClaimCenter 4.0.x, these fields live as extensions on `MobilePropertyIncident`.
- In ClaimCenter 5.0.0, Guidewire moved these fields into a separate `LocationAddress` object, in the core data model.
- In ClaimCenter 5.0.1, Guidewire changed `LocationAddress` into an extension on `MobilePropertyIncident` (in `extensions.xml`).

Upgrading from ClaimCenter 3.1.x or 4.0.x. Guidewire added a version check that should prevent data loss when upgrading to 5.0.2 from either 4.0.x or 3.1.x. It will look for the `LocationXXX` fields in incident or exposure (depending on whether you are upgrading from 4.0.x or 3.1.x). If any of those fields exist, then the version check will insist the following:

1. There is a `LocationAddress` field, of type `Address`, somewhere in the incident hierarchy
2. That any incidents/exposures with the `LocationXXX` fields set map to incident types that contain the `LocationAddress` field. That is, it checks that all `LocationXXX` fields that contain a value have a `LocationAddress` field to use in migrating the data.

If either of these checks fail, the version check will abort the upgrade and you must then follow the steps outlined in the workaround to solve the problem.

Upgrading from ClaimCenter 5.0.0. If you are upgrading from ClaimCenter 5.0.0, there is already a `LocationAddress` field. However, in 5.0.0, Guidewire placed the field in the core data model, while in 5.0.2 it is in `extensions.xml`. So, if your 5.0.2 `extensions.xml` does not contain a `LocationAddress` field, then the `LocationAddress` field could get dropped entirely during the upgrade process, possibly losing data. This is the case that the upgrade version trigger does not address.

Upgrading from ClaimCenter 5.0.1. This is not an issue.

Workaround: Depending on the release from which you are upgrading, you need to do the following:

- **ClaimCenter 3.1.x base release.** In 3.1.x, the `LocationXXX` fields were in the base data model, on exposure. As part of the upgrade, you will need to decide which exposure types map to which incident subtype. You must map any exposure types that have the `LocationXXX` fields set to an incident subtype that has the `LocationAddress` field. This ensures that the upgrade process has a place to move the data.
- **ClaimCenter 4.0.x base release.** In 4.0.x, the `LocationXXX` fields were in `extensions.xml`, in the incident hierarchy. You could choose where in the incident hierarchy to put these fields. In upgrading to any 5.0.x version, replace these fields in `extensions.xml` with `LocationAddress`, at the same level in the hierarchy.
- **ClaimCenter 5.0.0 base release.** In 5.0.0, the `LocationAddress` field was in the base data model, in `MobilePropertyIncident`. If you are upgrading to 5.0.x, you will need to make sure `LocationAddress` is in the `extensions.xml` file. Guidewire recommends that you put it in `MobilePropertyIncident`. However, if desired, it is possible—because it now lives in `extensions.xml`—to move it to a different place in the incident hierarchy.

Transaction `runWithNewBundle` Does Not Follow Links (CC-42654)

The `GScript` API that runs a block of code with new bundle (`gw.transaction.Transaction.runWithNewBundle(...)`) has some limitations. Even if you call `newBundle.add(yourentity)`, the `add` method does not follow the graph links to other entities to which it links. You must manually add each sub-object to the new bundle, using the call `bundle.add(linkedEntity)` for each entity. This includes all foreign key references to other entities as defined in the built-in data model files, but also includes customer data model extensions that include foreign key references.

You can work around this by committing all entities in the current bundle instead of using this API. This may be an issue when designing implementing a custom web service. Committing all entities in the current bundle includes all entities serialized into the bundle as part of the web service request because they were parameters to a web service method. There may be cases where entities that are web service API parameters you do **not** want commit to the database.

One workaround is to redesign the web service API so instead of taking entities (the entities you do not want to commit) it takes `GScript` class instances instead of Guidewire entities. Your custom `GScript` classes can provide only the fields that you want to pass in your integration code, which may be a much smaller subset of fields in some cases. Because they are `GScript` classes and not entities, committing the entire bundle with `gw.transaction.Transaction.getCurrent().commit()` will skip that data that you choose not to commit to the database.

Creating a new document from a template results in a `ClassNotFoundException` (CC-42887)

Issue: If you implement the `IDocumentProduction` plugin in Java, then when you create a new document from a template you may encounter a `ClassNotFoundException`.

Workaround: Copy the file `ClaimCenter/modules/cc/config/plugin/ExternalMappings.properties` to `ClaimCenter/modules/configuration/config/plugin`. Then add the following line to the copied file:

```
gw.api.document.DocumentCreationInfo = gw.api.document.DocumentCreationInfo
```


In addition, copy the file `ClaimCenter/modules/cc/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf` to `ClaimCenter/modules/configuration/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf`. In the copied file, replace the following line:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.DocumentTemplateDescriptor
      .TemplateId, gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo,
        DocumentCreationInfo.Document)))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

with the following:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    util.document.DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.
      DocumentTemplateDescriptor.TemplateId,
      gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo,
        DocumentCreationInfo.Document)))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```


Guidewire ClaimCenter 5.0.3 Release Notes

Release 5.0.3.17

Copyright © 2008 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire ContactCenter, Guidewire Insurance Suite, and the Guidewire logo are trademarks 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 documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 5.0.3.17.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact Guidewire Customer Support. You may contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For installation information, please refer to the *ClaimCenter Installation Guide*.

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Configuration Parameter Changes
- Improvements and General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 5.0.2 and ClaimCenter 5.0.3, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- `gw.api.*` GScript API
- typelists
- display keys

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.2 to 5.0.3

- 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/platform` directory, [click here](#).

Base Resource Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.2 to 5.0.3

- To view a report of the changes in the base resources in the modules/cc directory, [click here](#).
- To view a report of the changes in the base resources in the modules/core directory, [click here](#).
- To view a report of the changes in the base resources in the modules/platform directory, [click here](#).

Configuration Parameter Changes

The following are the primary changes to the ClaimCenter configuration parameter default settings specified in config.xml:

- Within the <database> element, the format of the archive subelements has changed. The old format was:

```
<param name="jdbcURLarchive" value="jdbc:h2:file:/tmp/guidewire/ccarchive"/>
<param name="jdbcURLarchivetest" value="jdbc:h2:file:/tmp/guidewire/ccarchivetest"/>
<param name="archive.reference" value="L"/>
```

The new format is:

```
<archive name="archive1" driver="dbcp" reference="L">
  <param name="jdbcURL" value="jdbc:h2:file:/tmp/guidewire/ccarchive"/>
  <param name="jdbcURLtest" value="jdbc:h2:file:/tmp/guidewire/ccarchivetest"/>
</archive>
```

- Added new parameter ArchiveEnabled with default value of true.
- Added new parameter DiscardQueryPlansDuringStatsUpdateBatch with default value of false.

Improvements and General Issues

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

ID	Description
Administration	
CC-45703	Fixed an issue where running regen-dictionary would display unnecessary warnings.
Archiving	
CC-44805	Fixed an issue in which the Retrieve button was enabled when the archive database was not available.
CC-45290	Implemented a database schema for archive in config.xml. There is now an optional <archive> element under <database>, with attributes name, driver, reference, and passwordFile. It also has <param> children, for specifying the JDBC URL, plus other connection parameters.
CC-46535	Added missing entries for the "markforarchive" and "archive" batch processes in the processtype enumeration in the schema for scheduler-config.xml (scheduler-config.xsd). The missing entries caused an error during scheduler startup.
Assignment / Segmentation / Strategy	
CC-43473	When an assignment changes, there is an AssignmentAdded, AssignmentChanged, or AssignmentRemoved event. See "Assignment Events" on page 48 in the Rules Guide for details.
Clustering and related issues	
CC-45485	Implemented steps to prevent a server from joining the cluster if it has a different configuration from the cluster.
Configuration Upgrade	
CC-44608	Corrected an issue in which the base application rules and the customized rules were not handled separately and correctly during upgrade.
ContactCenter Integration	
CC-44956	Corrected an issue with Contact linking code which resulted in performance problems, and ultimately an out-of-memory error.
Database Upgrader	
CC-44809	Fixed an error that would occur when upgrading from 3.1 with camel-case extensions to the cc_bodypart table.
CC-45030	Fixed an issue that caused the database upgrader to fail if the default application locale was set to a non-English locale.

ID	Description
CC-45285	Fixed an error that would cause a database upgrade to fail if any of the 'Purge', 'STPopulate', or 'STValidate' process history types exist in the processhistory table.
Documentation and help	
CC-45710	When using Sun JVM with WebLogic, add the -server flag as an argument when launching the WebLogic start script. See the ClaimCenter Installation Guide for details.
CC-46100	<p>Creating a SQL Server database with files of sufficient size and parameters is important to its future performance and maintenance. A basic discussion of this can be found online in a Microsoft SQL Server topic "Designing Databases" at ms-help://MS.SQLCC.v9/MS.SQLSVR.v9.en/udb9/html/860e930d-ed5c-42c2-a384-28756cb16176.htm.</p> <p>Consider the use of filegroups to physically distribute the database over separate spindles. Estimate how big your database will grow in one year and add 20%, then allocate enough total file space for this size.</p> <p>It is better to monitor the size of the database and add space during scheduled off-hours than to use auto-growth, but you may consider setting autogrowth to avoid running out of space in the database, particularly test databases. Set the maximum file size to be less than the size of the disk, so that the disk does not fill up.</p> <p>For your production database, work with your SAN engineers to deliver them performance realistic requirements early in the cycle.</p> <p>SQL Server users should also be aware that INDEX is a reserved name and therefore can not be used for a physical filegroup name.</p>
Exposures	
CC-44781	In 5.0.0, LocationAddress was part of the core data model, in MobilePropertyIncident. Starting in 5.0.1, it was moved it to extensions.xml. If you are upgrading from 5.0.0 -> 5.0.x you are now prevented from upgrading unless there is a LocationAddress field somewhere in the Incident hierarchy.
CC-46179	Fixed an issue where exposures added to a Workers' Comp claim after the completion of the New Claim wizard would be added at the 'Draft' status.
Financials	
CC-44176	Improved the performance of the list view query in ClaimFinancialsTransactions.pcf.
CC-44448	Fixed an issue where a bulk invoice in "In Review" status would not get changed to Draft when one of its items' reserve lines changed.
CC-44861	Calling IClaimFinancials.updateCheckStatus() from a messaging plugin will not work in 5.0.x. The new method Check.updateCheckStatus() should be called instead, but only for the cases in which IClaimFinancials.updateCheckStatus() was appropriate (i.e. "pendingvoid" -> "voided"). Most status transitions still occur through methods on the Message interface (i.e. submittingCheck()).
CC-45011	Fixed an issue where in the "By Reserve Lines" list view, when the "GlobalUtilizeRecoveryReserves" script parameter was set to true, if there were no Subrogation Recoveries but there were Open Recovery Reserves for a Reserve Line, the Anticipated Recovery % cell incorrectly showed 0% for that row.
GScript language	
CC-44673	Fixed an issue that would occasionally cause an exception during evaluation of an if-then statement.
CC-45009	Corrected an issue in which null function return values were not converted to default values when cast to a primitive.
Internationalization	
CC-40589	<p>Due to an issue in which ClaimCenter did not properly display localized typelists in localized reports, you now need to run one of the following scripts whenever you add either a new typelist or a new locale to your environment. Choose the script depending on your database:</p> <ul style="list-style-type: none"> - cc_oracle_views.sql - cc_sqlserver_views.sql <p>You can find these scripts in your reporting folder (not the ClaimCenter application folder) under resources.</p>
CC-44572	Fixed an issue where the Parameter Claim Status would not reflect the user locale setting on the Claim Detail - LOB drill down report.
CC-44574	Fixed an issue where report parameter labels would be truncated on the Claim Catastrophe Detail report for non-English locales.

ID	Description
CC-44576	Fixed an issue where the "All" default parameter value for reports would not be correctly localised in non-English locales.
CC-44577	Fixed an issue where the Loss Cause value in a report was not correct when run in a non-English locale.
CC-44578	Fixed an issue where running a report with the Loss Cause parameter set and a non-English locale would result in no data appearing in the report.
CC-46270	Fixed an issue where drill-down reports would not return any data for locales other than English US.
New Claim Wizard	
CC-44942	Fixed an error that would occur when choosing View Details for a vehicle that was not on a policy.
CC-45055	Fixed an issue where the Notes page in the New Claim wizard was missing the Back button.
CC-45162	Fixed an issue where the base catastrophe validation rule would incorrectly determine that the Catastrophe field had not changed.
CC-45239	Fixed an exception that would occur when navigating to the ClaimSummary screen from the FNOL wizard.
CC-45304	Fixed an error that would occur when setting an insured name, changing the policy, and then adding back the same insured name.
CC-45332	Added correct IDs to the Choice PCF widget in the New Claim wizard.
CC-45351	Fixed an error that would occur after clicking Go to Full Wizard in the Quick Claim Property wizard.
CC-45437	Fixed an issue where you may be unable to enter the loss location information at a step 2 of 2 in the GL, Incident Only New Claim wizard with an unverified policy selected.
CC-45466	Added correct IDs to the Choice PCF widget in the GL New Claim wizard.
Performance	
CC-46536	An index claimu7u will now be used by queries providing data for the Team Group Activities page and several other pages involving claimnumber like Desktop Activities. This index was original designed to address claim searches using losssdate and will continue to serve that purpose. If these are not used in your configuration, you can remove this index from extensions.xml and cause an upgrade.
Platform	
CC-45326	Some upgrade backup SQL Server databases were created in SQL Server 2000. When they are restored to a more recent version of SQL Server, they will be converted but left at a compatibility_level of 80 for SQL Server 2000. This needs to be set to be at 90 for SQL Server 2005 or 100 for SQL Server 2008.
PolicyCenter Integration	
CC-44224	Fixed an issue where the address would appear twice in the Location drop-down in step 2 of the New Claim wizard, Quick Auto Claim.
CC-44300	Fixed an issue where Drivers and Secondary Named Insured contacts would not transfer from PolicyCenter to ClaimCenter.
CC-44627	Fixed an issue where Excluded Parties on a PolicyCenter policy would not be reflected on the New Claim wizard Policy General page in ClaimCenter.
CC-44629	Fixed an issue where a null limit would be displayed for Rental Coverage.
CC-44996	Fixed an issue where the endorsement number displayed in the comments field for a workers' comp policy would be null if integrating with PolicyCenter.
Server issue	
CC-45238	Corrected an issue that allowed the application server to start with no exception even if the gscriptclass attribute in a plugin.xml file was set to a non-existent class. The application server now verifies that the implementation types exist as part of plugin configuration startup.
CC-45575	Correct an issue for which the Total Cluster wait time showed the wrong unit-value mapping in AWR reports of RAC.

ID	Description
CC-46133	<p>To address database performance on Oracle, the following new configuration parameters are available:</p> <pre> <!-- ClaimCenter works around Oracle bug 5886252 by disabling index fast full scan when executing certain claim searches on Oracle. This parameter controls the work around and is true by default. If a future version of Oracle fixes the defect this parameter may be removed. The parameter has no effect on databases other than Oracle. --> <param name="DisableIndexFastFullScanForClaimSearch" value="true"/> <!-- ClaimCenter works around Oracle bug 6990305 by disabling optimizer cost base transformation when executing certain claim searches on Oracle. This parameter along with DisableIndexFastFullScan- ForClaimSearch controls the work around and is true by default. If a future version of Oracle fixes the defect this parameter may be removed. The parameter has no effect on databases other than Oracle --> <param name="DisableCBQTFForClaimSearch" value="true"/> <!-- ClaimCenter works around hash join related query plan problems when executing certain claim searches on Oracle. This parameter controls part of the work around and is true by default. The parameter has no effect on databases other than Oracle --> <param name="DisableHashJoinForClaimSearch" value="true"/> <!-- ClaimCenter works around sort merge join query plan problems when executing certain claim searches on Oracle. This parameter controls part of the workaround when DisableHashJoin- ForClaimSearch is set to true and is true by default. The parameter has no effect on databases other than Oracle --> <param name="DisableSortMergeJoinForClaimSearch" value="true"/> <!-- ClaimCenter works around semi join query plan problems by forcing nested loop semi join instead of choose when executing certain claim searches on Oracle. This parameter controls part of the work around and is true by default. The parameter has no effect on databases other than Oracle --> <param name="SetSemiJoinNestedLoopsForClaimSearch" value="true"/> </pre>
SOAP APIs	
CC-45133	Changed the names of objects used in the PolicyCenter-ClaimCenter integration web service so they do not clash with other SOAP objects.
Studio	
CC-44470	Fixed several Studio Rule Debugger issues.
CC-44523	Corrected an issue in which Studio did not create a correct and functional Web Service end-point proxy.
CC-44706	Fixed an issue in which the Studio Web Service tool did not correctly handle numerical enumeration datatypes in imported .wsdl files.
CC-45429	In previous versions of Studio, you could not save a change to a Display Key in the customer build if the key was in the Web, Quickjump, ProductModel, PolicyPeriod, Diff, Global, Java, or Job directory. This has been fixed.
Subrogation	

ID	Description
CC-45043	<p>Fixed the following issues with subrogation financials:</p> <ul style="list-style-type: none"> -The "Set Open Recovery Reserve to Expected Recovery %" button is now inactive if the Expected Recovery % equals the Anticipated Recovery %. -The algorithm to allocate the appropriate Recovery Reserve previously iterated through all ReserveLines with the following criteria: <code>CostType == "claimcost" and getNetPaidExcludingSubroRecovery() > 0</code> The criteria has changed to: <code>CostType == "claimcost" and (getNetPaidExcludingSubroRecovery() > 0 or FinancialsCalculationUtil.getOpenRecoveryReserves().getAmount(eachRL) > 0)</code>
Web	
CC-44529	If a modal PCF page has no default mode defined and the mode cannot be determined at runtime, an error is now shown.
CC-44808	Fixed an issue that caused PCF verification to fail when the entry point parameter on a PCF was an array of generic type.
Workflow	
CC-44757	Fixed a workflow issue that allowed an empty string to evaluate to true in a Trigger branch's permission or available conditions, and in a Go branch condition. Studio now displays errors during verification when these fields only have comments. Also fixed an issue with duplicate error messages during verification of the workflow. If the Go branch in an AutoStep has an error, that error was shown twice.
CC-44894	Corrected an issue in which a validation failure for a workflow Autostep's Go branch appeared twice in verification results.

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER/Software/Microsoft/Windows/CurrentVersion/Explorer/BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of #####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Using a message handling plugin to deny a check containing a final payment causes an error (CC-27165)

Issue: Using a message handling plugin to deny a check containing one or more 'final' payments causes an error. Denying checks that contain only 'partial' and 'supplemental' payments via a message handling plugin works correctly.

Workaround: Use the `IClaimFinancialsAPI` to deny a check with a final payment instead of a message handling plugin. Guidewire may fix this issue in a future release.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short may result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either `ContactCenter` or `ClaimCenter`) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of `ContactCenter` and to customers newly deploying Geocoding on an existing `ContactCenter` server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer

manually using the **System Tools** page, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Changing a coverage now fires the *Policy_Changed* event in addition to the *Coverage_Changed* event (CC-29564)

Issue: Changing a field on a Coverage now fires the *Policy_Changed* event, in addition to the *Coverage_Changed* event that has always fired.

Workaround: None.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the *ab_abaddress* table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` method in your rules, and the *ExposureType* that you pass only maps to a single *LossPartyType*, then that *LossPartyType* will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no *PrimaryCoverage* or *LossParty*.

Workaround: You should add a Pre-update rule that sets the *PrimaryCoverage* and *LossParty* fields to a non-null value.

IClaimAPI depends on *SynchStateData*, which uses deprecated methods (CC-31611)

Issue: *IClaimAPI* depends on *SynchStateData*, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

An exception is thrown when trying to link a new contact to the address book from the new check wizard (CC-32732)

Issue: If you create a new contact in the New Check Wizard, and later attempt to link contacts to ContactCenter, you get an error.

Workaround: None.

Imported voided/stopped checks have a \$0 gross amount (CC-33755)

Issue: If you import voided or stopped checks, then those checks appear on the **Checks** screen with a gross amount of \$0. The gross amount should actually be the original pre-voided/stopped amount.

Workaround: Run the Financial Calculations batch process after import.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To retrieve the payment's send date, both the Transactions and the Checks table would have to be searched; this would hamper performance and is not done.

Workaround: None.

The PCF Converter does not handle column removal from a column group (CC-34512)

Issue: Any customization of an NVV/LV file that removes a column from a column group causes the PCF converter to fail, and prevents upgrading to 4.0.x.

Workaround: You must comment out the change that removes columns, then rerun the PCF Converter.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The Reset button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the Reset button an Initialize button.

After logout, a request to return to the login page creates a new session in Websphere (CC-36394)

Issue: On logout in WebSphere, the current session is destroyed but a new one is created. In ClaimCenter, logging out takes you back to the login page. Whenever a browser interacts with an application server, the server creates a session if none existed.

Workaround: On logout, go outside of the Guidewire application and enforce static content caching: the browser will reuse static objects in its cache (like the login page) and therefore minimize the unnecessary interactions with the application server. Alternately, have a web tier cache the static content. This is basically adding another caching layer (on top of the browser) and would reduce this phenomenon further.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you go to the ClaimCenter Desktop Activities page and repeatedly click Activities, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to Internet Explorer 7.

regen-toolkit results in a "code too large" error (CC-37415)

Issue: When you run 'regen-toolkit', you receive an error that says 'code too large' when it is processing a typelist. This occurs when you have a typelist that is too large.

Workaround: Reduce the size of the typelist. The typelist size depends on the number of typekeys and the length of the strings (code, name, description) used to define them. You can reduce the typelist size by reducing any of these (for example, you can include more typekeys by shortening the description text).

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the Plan of Action > New Evaluation page will display an incorrect title ("New Evaluation" instead of "LOBName Evaluation"), and the page will not have a "LOBName Cost" field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use `ClaimEvaluationDetail` page instead of `ClaimEvaluationDetails<LOB>` page. Also, do not customize the `ClaimEvaluationDetail` page heavily.

Acrobat Sample document template does not allow extra fields (CC-40462)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

FNOL wizard's Basic Info page has incorrect *post on change* behaviors (CC-40479)

Issue: On the **Basic Info** (second) page of the New Claim wizard, editing the **Reported By** contact and then switching to another contact can sometimes edit the wrong contact and can also commit multiple contact changes.

Workaround: None.

You cannot delete a workflow, and an invalid workflow prevents the server from starting (CC-40533)

Issue: It is not possible to delete a workflow. Further, if you invalidate a workflow in the process of editing it, the invalidated workflow will prevent the application server from (re)starting.

Workaround: Instead of deleting an obsolete workflow, make sure that your application never triggers it. Also, use Studio's workflow editor to make sure all workflows are valid.

After upgrading from ClaimCenter 3.1, the doctor is no longer shown in medical detail exposures (CC-40595)

Issue: ClaimCenter 3.1 displayed the value `Exposure.Doctor`, but later versions of ClaimCenter display the value `Claim.FirstIntakeDoctor`. During upgrade, the data is not migrated from the old field to the new field, resulting in the upgraded ClaimCenter version showing a blank instead of the real value.

Workaround: Add another PCF field in ClaimCenter 5.0 on the **Medical Details** page below the current **Doctor** field for `Exposure.Claim.FirstIntakeDoctor`. This will display the upgraded data.

ClaimCenter uses an old version of Castor libraries (CC-40659)

Issue: Guidewire products still use an old version of the Castor library, and you cannot modify the version used. The version in ClaimCenter is bundled version 0.9.4.3. The current version of the Castor libraries is 1.2.

Workaround: You may use another version of this library within your plugin code. To use a different version of Castor, add its jar to the `modules/configuration/plugins/<plugindir>/lib` folder. All jar files located in the plugins directory are loaded with a different class loader than the one used by the ClaimCenter core code. This avoids any potential conflicts between versions of the libraries.

Potential to lose LocationXXX data during upgrade (CC-41555)

Issue: Guidewire changed the location of the `LocationCity`, `LocationState`, `LocationStreet`, and `LocationZip` fields between ClaimCenter 3.1.x, 4.0.x, 5.0.0, and 5.0.1:

- In ClaimCenter 3.1.x, these fields live on the exposure.
- In ClaimCenter 4.0.x, these fields live as extensions on `MobilePropertyIncident`.
- In ClaimCenter 5.0.0, Guidewire moved these fields into a separate `LocationAddress` object, in the core data model.

- In ClaimCenter 5.0.1, Guidewire changed `LocationAddress` into an extension on `MobilePropertyIncident` (in `extensions.xml`).

Upgrading from ClaimCenter 3.1.x or 4.0.x. Guidewire added a version check that should prevent data loss when upgrading to 5.0.2 from either 4.0.x or 3.1.x. It will look for the `LocationXXX` fields in incident or exposure (depending on whether you are upgrading from 4.0.x or 3.1.x). If any of those fields exist, then the version check will insist the following:

1. There is a `LocationAddress` field, of type `Address`, somewhere in the incident hierarchy
2. That any incidents/exposures with the `LocationXXX` fields set map to incident types that contain the `LocationAddress` field. That is, it checks that all `LocationXXX` fields that contain a value have a `LocationAddress` field to use in migrating the data.

If either of these checks fail, the version check will abort the upgrade and you must then follow the steps outlined in the workaround to solve the problem.

Upgrading from ClaimCenter 5.0.0. If you are upgrading from ClaimCenter 5.0.0, there is already a `LocationAddress` field. However, in 5.0.0, Guidewire placed the field in the core data model, while in 5.0.1 and later it is in `extensions.xml`. So, if your target version `extensions.xml` does not contain a `LocationAddress` field, then the `LocationAddress` field could get dropped entirely during the upgrade process, possibly losing data. This is the case that the upgrade version trigger does not address.

Upgrading from ClaimCenter 5.0.1 or later. This is not an issue.

Workaround: Depending on the release from which you are upgrading, you need to do the following:

- **ClaimCenter 3.1.x base release.** In 3.1.x, the `LocationXXX` fields were in the base data model, on exposure. As part of the upgrade, you will need to decide which exposure types map to which incident subtype. You must map any exposure types that have the `LocationXXX` fields set to an incident subtype that has the `LocationAddress` field. This ensures that the upgrade process has a place to move the data.
- **ClaimCenter 4.0.x base release.** In 4.0.x, the `LocationXXX` fields were in `extensions.xml`, in the incident hierarchy. You could choose where in the incident hierarchy to put these fields. In upgrading to any 5.0.x version, replace these fields in `extensions.xml` with `LocationAddress`, at the same level in the hierarchy.
- **ClaimCenter 5.0.0 base release.** In 5.0.0, the `LocationAddress` field was in the base data model, in `MobilePropertyIncident`. If you are upgrading to 5.0.x, you will need to make sure `LocationAddress` is in the `extensions.xml` file. Guidewire recommends that you put it in `MobilePropertyIncident`. However, if desired, it is possible—because it now lives in `extensions.xml`—to move it to a different place in the incident hierarchy.

Transaction `runWithNewBundle` Does Not Follow Links (CC-42654)

The `GScript` API that runs a block of code with new bundle (`gw.transaction.Transaction.runWithNewBundle(...)`) has some limitations. Even if you call `newBundle.add(yourentity)`, the `add` method does not follow the graph links to other entities to which it links. You must manually add each sub-object to the new bundle, using the call `bundle.add(linkedEntity)` for each entity. This includes all foreign key references to other entities as defined in the built-in data model files, but also includes customer data model extensions that include foreign key references.

You can work around this by committing all entities in the current bundle instead of using this API. This may be an issue when designing implementing a custom web service. Committing all entities in the current bundle includes all entities serialized into the bundle as part of the web service request because they were parameters to a web service method. There may be cases where entities that are web service API parameters you do **not** want commit to the database.

One workaround is to redesign the web service API so instead of taking entities (the entities you do not want to commit) it takes `GScript` class instances instead of Guidewire entities. Your custom `GScript` classes can provide only the fields that you want to pass in your integration code, which may be a much smaller subset of fields in some cases. Because they are `GScript` classes and not entities, committing the entire bundle with

`gw.transaction.Transaction.getCurrent().commit()` will skip that data that you choose not to commit to the database.

Creating a new document from a template results in a `ClassNotFoundException` (CC-42867)

Issue: If you implement the `IDocumentProduction` plugin in Java, then when you create a new document from a template you may encounter a `ClassNotFoundException`.

Workaround: Copy the file `ClaimCenter/modules/cc/config/plugin/ExternalMappings.properties` to `ClaimCenter/modules/configuration/config/plugin`. Then add the following line to the copied file:

```
gw.api.document.DocumentCreationInfo = gw.api.document.DocumentCreationInfo
```

In addition, copy the file `ClaimCenter/modules/cc/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf` to `ClaimCenter/modules/configuration/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf`. In the copied file, replace the following line:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.DocumentTemplateDescriptor
      .TemplateId, gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

with the following:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    util.document.DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.
      DocumentTemplateDescriptor.TemplateId,
      gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

After upgrade, two duplicate incidents are created (CC-44646)

Issue: If you have two exposures with the same incident type details (for example, auto damage on the same car with the same driver), the database upgrader creates two identical incidents.

Workaround: Please contact Guidewire Support for assistance.

Type name conflict in web services (CC-45132)

Issue: It is illegal to have two types with the same relative name within the context of the same web service, or within the context of any published web service.

Workaround: Use unique names.

Limitation in reporting when a typelist name contains a comma (CC-45305)

Issue: Due to the limitations of the InetSoft reporting software, issues arise when a passing a typelist as a parameter to a drill-down report if one of its typecode values contains a comma. (For example, typelist Loss Cause contains typecode "Fall, slip, or trip injury".) If a typecode does contain a comma, InetSoft thinks it is an array and sends multiple values. Guidewire has implemented a solution for the Loss Cause typelist in ClaimCenter. However, if you are passing any other typelist as a parameter to a drill-down report and it contains a typecode with a comma, you will experience this issue.

Workaround: Guidewire is aware of this issue and will address it in a future release. If you encounter this issue with any other report parameters, please contact Guidewire Support for assistance.

Cannot upgrade to ClaimCenter 5.0.3 if you implemented archiving in an earlier release (CC-45993)

Issue: If you are using the archiving feature in ClaimCenter 5.0.0 through 5.0.2, you cannot upgrade to ClaimCenter 5.0.3.

Workaround: Please contact Guidewire Support for assistance. If you start using archiving in ClaimCenter 5.0.3, there will be no issues with future upgrades.

Typelist error message: "contains a typecode with an empty code/name" (CC-46085)

Issue: When starting the server, you may see an error that says a typelist contains a typecode with an empty code or name. This error is mistakenly generated and does not indicate any problem.

Workaround: Ignore this error.

ContactCenter error when changing the LinkID or PublicID (CC-46392)

Issue: When updating contacts via the ClaimCenter-ContactCenter integration, ContactCenter assumes that the contact (and its related entities, including Address) has a PublicID equal to the LinkID (exposed as AddressBookUID through the integration on ClaimCenter). This is necessary to ensure that the contact is properly updated. If you later modify the PublicID, the contact will not be updateable from ClaimCenter. This can also happen by changing the LinkID to a different value when importing contacts directly into ContactCenter through staging tables.

Workaround: If you encounter this situation, please contact Guidewire Customer Support for assistance.

Upgrade fails when upgrading CoverageType typelist with value BLDG (CC-46510)

Issue: During upgrade, you receive an exception for the following upgrade trigger: RecodeCoverageType-codesTriggers\$RecodeCoverageTypecode

Workaround: Please contact Guidewire Support for assistance.

A PCF page with the startEditing attribute will not commit (CC-46740)

Issue: If you define a PCF page with the startEditing attribute defined, the page will not commit when you try to edit and save it.

Workaround: Guidewire is aware of this issue and will address it in a future release.

chapter 28

Guidewire ClaimCenter 5.0.4 Release Notes

Release 5.0.4.13

Copyright © 2001-2009 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 5.0.4.13.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, please contact Guidewire Customer Support. You may contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For general installation information, refer to the *ClaimCenter Installation Guide*. If you plan to use the integrated reporting feature, refer to the *ClaimCenter Reporting Guide* for details.

The following issues pertain to installing this release.

Inetsoft Evaluation

You can evaluate Inetsoft Style Report Enterprise Edition Version 9.0 using the following license key:

L000-69C-ERX-000097000006-F2A7343ACB01

If you want to use Inetsoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide*.

Wrapping of Labels

This release includes a change (PL-271) to improve user interface performance. This change will likely have a noticeable effect on the spacing of some elements on the ClaimCenter screens. In particular, Internet Explorer's rendering of HTML tables may cause labels to wrap more aggressively than in previous ClaimCenter versions. The wrapping occurs most commonly in the left-most columns of detail views, but might appear in other places throughout the product interface.

After upgrading to this release, Guidewire recommends that you look through the user interface of your installation to identify places where labels wrap in undesirable ways. There are a few ways to control how the browser renders and wraps ClaimCenter labels. For further details, refer to *this document*. (Note: This link requires the `readme_files` directory on your local disk.)

Method Removed

Guidewire has removed the `Collection.toMap()` method due to an incompatibility with the LINQ specification. Guidewire reserves the right at a later time to implement a method with this same name but different behavior to match industry conventions for this method name. For upgrade examples, refer to the Upgrade Guide.

Changes to Parameter Order in Outgoing SOAP API Requests

In this release, Guidewire changed how the GScript language accesses remote web services registered in Studio. Due to this fix (CC-50027), for some remote SOAP API methods, the parameter order of APIs might change from the perspective of the GScript SOAP client code. ClaimCenter now fixes parameter order so it always uses the parameter order in the method signature in the WSDL. To avoid unexpected behaviors upgrading from previous releases, Guidewire provides a new configuration setting. However, to use the old behavior, set the configuration parameter `SortWSDLsUponStudioImport` in the `config.xml` file to the value `true`. This value enforces backward compatibility for the previous release of ClaimCenter.

If you set this parameter to `false`, the order of arguments in some cases change for method signatures calls to remote SOAP API methods. Review all outgoing SOAP API calls to confirm the correct parameter order before the next major release. However, you can defer this work until the next major upgrade by using the new configuration parameter.

WARNING Not all affected methods can be detected from compile errors. For example, if a method takes two instances of the same type as function arguments. If the order changes, then there is no compile error.

Setting `SortWSDLsUponStudioImport` to `true` fixes a compile error in the included MapPoint implementation. If you set this parameter to `false`, then see the following section about MapPoint.

If you are *not* upgrading from a previous release of ClaimCenter, then Guidewire recommends that you disable this setting (set it to `false`). This ensures that no conflicts occur with later major upgrades. This might require additional configuration work. If you set the new configuration parameter `SortWSDLsUponStudioImport` in the `config.xml` file to the value `false`, then review all your code that calls to remote SOAP APIs from GScript to confirm whether parameter orders changed for any methods.

Studio updates types and API method signatures when it loads or refreshes the WSDL. To refresh the WSDL, click **Refresh** or **Edit** in the Studio Webservices editor.

To change the WSDL parameter order setting:

1. Set the `SortWSDLsUponStudioImport` parameter in the `config.xml` file. If you are upgrading from a previous release, then set to `true` for maximum backward compatibility. Set to `false` if you are a new project, or you are upgrading from a release earlier than ClaimCenter 5.0.
2. Open Studio. Access each web service and click **Refresh**.
3. Verify the correctness of that web service:
 - a. In Studio, run the **Verify All Resources** command.
 - b. Search for usages of the API stub class corresponding to that particular web service.
 - c. Inspect each operation web service method call to ensure that method parameters are in the correct order. As mentioned before, not all affected methods can be detected from compile errors.

MapPoint Updates

In this release, Guidewire changed how the GScript language accesses remote web services registered in Studio. The included MapPoint implementation that calls to the remote web service now works with either value of the new configuration parameter `SortWSDLsUponStudioImport`.

If you set `SortWSDLsUponStudioImport` to `true`, which is recommended for upgrades, do not make changes to the included MapPoint implementation.

If you want to use the correct (fixed) parameter order, set the value of `SortWSDLsUponStudioImport` to `false`. You get compile errors within MapPoint code, but you can update MapPoint to work with this setting. To update the MapPoint implementation to work with the fixed parameter order (that is, with `SortWSDLsUponStudioImport` set to `false`). To do so:

1. Set the `SortWSDLsUponStudioImport` parameter in the `config.xml` file to `false`.
2. Open the Studio Webservices editor and select the *MapPoint* web service.
3. Click **Refresh**.
4. Navigate to the GScript class `gw.plugin.geocode.impl.MappointGeocodePlugin`.

5. Modify line 175 to:

```
var route = routeService.CalculateSimpleRoute ( startAndFinish, dataSourceName, segPref );
```

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Configuration Parameter Changes
- Improvements and General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 5.0.3 and ClaimCenter 5.0.4, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- `gw.api.*` GScript API
- typelists
- display keys

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.3 to 5.0.4

- 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/platform` directory, [click here](#).

Base Resource Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.3 to 5.0.4

- To view a report of the changes in the base resources in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base resources in the `modules/core` directory, [click here](#).
- To view a report of the changes in the base resources in the `modules/platform` directory, [click here](#).

Configuration Parameter Changes

The following are the primary changes to the ClaimCenter configuration parameter default settings specified in `config.xml`:

- Added the following new parameters:
 - `DisableIndexFastFullScanForTeamGroupActivities` — ClaimCenter works around index fast full scan related query plan problems when executing the team group activities page’s main query on Oracle. This

parameter controls the workaround and is true by default. If a future version of Oracle fixes the defect, this parameter may be removed. The parameter has no effect on databases other than Oracle.

- **DisableCBQTFForTeamGroupActivities** — ClaimCenter works around optimizer cost base transformation related query plan problems when executing the team group activities page's main query on Oracle. This parameter controls the workaround and is true by default. If a future version of Oracle fixes the defect, this parameter may be removed. The parameter has no effect on databases other than Oracle.
- **DisableHashJoinForTeamGroupActivities** — ClaimCenter works around hash join related query plan problems when executing the team group activities page's main query on Oracle. This parameter controls part of the workaround and is true by default. The parameter has no effect on databases other than Oracle.
- **DisableSortMergeJoinForTeamGroupActivities** — ClaimCenter works around sort merge join query plan problems when executing the team group activities page's main query on Oracle. This parameter controls part of the workaround when **DisableHashJoinForClaimSearch** is set to true and is true by default. The parameter has no effect on databases other than Oracle.

Improvements and General Issues

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

ID	Description
Activities	
CC-47103	Fixed an issue where activities that should appear on the calendar were not displaying on the months where the first day of the month began with Monday.
CC-49052	Corrected an issue where you were previously unable to set the ReLatedTo property on an Activity using rules.
Administration	
CC-47561	Fixed a performance issue where viewing a user in the Administration screens could pull all the transactions they ever created into memory.
Application Framework	
PL-4093	Fixed an issue with the assignUserByLocationUsingProximityAndAttributes assignment method that incorrectly returned a less-than-complete set of users based on the proximity search, which then incorrectly failed to assign using the attribute filter due to the pool of users not being complete.
PL-4094	Fixed an issue with the assignUserByLocationUsingProximityAndAttributes assignment method that permitted assignment of an entity marked as requiring special handling to a user without the requisite permission.
Archiving	
CC-46497	The DaysClosedBeforeArchive configuration parameter, which specifies the number of days after a claim is closed before being considered for archiving, is now set to 30 days.
Claim File	
CC-45915	Fixed an issue when assigning a Claim to a particular Group, where the Group received a null bundle error. The modified AssignmentGroupLV.pcf file is accessed in Studio under resources > config > web > pcf > shared > assignment .
CC-46269	Fixed an issue where the visibility expression in the Claim menu would sometimes cause a null bundle error.
Configuration Changes	
CC-45299	An issue was fixed where a PCF panel was not used in the correct mode. This could result in incorrect exchange rates on a check with multiple payments. All payments on a check must point to the same ExchangeRate entity. The affected file in Studio: resources/config/web/pcf/claim/newtransaction/shared/NewPaymentDetailDV.pcf .
CC-48178	When a property policy was defined with a propertyRU with location-based coverages, the display name shown in the cascading New Exposure menu displayed the policy location ID value instead of a display name. A complete definition for the Display Name of the LocationBasedRU entity was added to the LocationBasedRU.xml file. This file is located in the Studio config > displaynames directory.

ID	Description
ContactCenter Integration	
CC-46392	Fixed an issue where updates from ClaimCenter to ContactCenter for a given Contact would fail if the LinkID and PublicID for that contact in ContactCenter did not match.
CC-47250	Fixed an issue where ClaimCenter could not display the details of a contact created in ContactCenter.
CC-47853	Fixed an issue where creating a new contact with a primary contact set through the ClaimCenter-ContactCenter integration failed.
Core	
PL-314	Added the @gw.lang.Exportable annotation. This annotation, if added to an entity enhancement property, causes the application server to include that property in SOAP messages. You must define your own getter and setter methods on the property for this to work properly.
PL-317	Fixed an issue in which a <i>WebServices Service Lookup</i> failed on a WSDL with multiple port protocols defined.
PL-4055	Fixed a performance issue that caused the application server to hang for up to several hours. This occurred if Cancel was clicked in multiple New Claim wizards simultaneously.
PL-4060	Fixed an issues that caused the Data Dictionary to incorrectly display unicode characters.
Database	
CC-45603	Fixed the ClaimantFlag update trigger.
CC-47600	Added a version check to fail early during upgrade, if the creation of the unique index on RiskUnit fails.
PL-332	Added a new configuration parameter, DbmsCounterThresholdArchiveSecs, that sets the batch archive threshold in seconds for the archive batch process. The default is 600 (seconds). To turn off this parameter, set it to -1.
PL-334	Fixed an issue that caused command <code>system_tools -password gw -checkdbconsistency</code> to fail if you specified any arguments, and the first argument was not "all", and if you had a typelist table among the tables specified.
PL-338	Guidewire now encodes file names in ZIP files as 7-bit ASCII, rather than UTF-8. This mitigates a problem with the unzipping process that incorrectly rendered these file names.
PL-346	Added the following to Info Tools Database Parameters: <ul style="list-style-type: none"> • Guidewire Archive Database Config • Archive Database Connection Properties • Archive Database Connection Pool Settings
PL-1819	Fixed an issue with the DBAuthentication plugin that incorrectly handled authentication of a user name and password read from a file.
PL-4092	Fixed an issue that caused an the application server log to print an incorrect database URL at system startup.
Documentation	
CC-50007	The GScript variable scope called <i>request</i> makes no sense outside a web sessions. Request scoped variables must never be triggered by code in a batch process. Attempting to do so results in server exceptions after an upgrade. Guidewire strongly recommends avoiding using request and session variables altogether in all cases, not just in code that could indirectly be triggered by batch processes.
PL-2438	In general, avoid calling locally-hosted SOAP APIs from within a plugin or the rules engine. Be careful about any SOAP calls to the same server. If the SOAP API hosted locally modifies entity data and commits the bundle, the current transaction does not always detect and reload local data. Instead, refactor your code to avoid this case. For example, write a GScript class that performs a similar function as the web service but that does not commit the bundle. This type of refactoring also results in higher server performance. If you have questions about how to convert some particular use locally-hosted SOAP APIs from plugins or rules, contact Customer Support. This is true for all types of local loopback SOAP calls to the same server: the soap.local.* GScript APIs, the Java toolkit SOAP APIs, and any Studio-registered web services that call the same server as the client.
Documents	

ID	Description
CC-45862	To improve document management performance, ClaimCenter has changed the file structure for document storage. New documents created after upgrading will use this new structure. However the location of existing documents will not change. If you are having performance issues with the old document structure and would like to migrate your existing documents to the new structure, contact Guidewire Customer Support for assistance.
Financials	
CC-45299	An issue was fixed where a PCF panel was not used in the correct mode. This could result in incorrect exchange rates on a check with multiple payments and now works correctly. All payments on a check must point to the same ExchangeRate entity. Affect file is at the path: config/web/pcf/claim/newtransaction/shared/NewPaymentDetailDV.pcf
CC-46328	Corrected an issue where you could not delete the last check out of a recurrence schedule.
CC-47111	Corrected an issue where you were unable to void a transferred check, when the system was configured to disallow multiple payments on a check (AllowMultiplePayments=false).
CC-47147	Corrected an issue where a check was created through QuickCheck and a bundle rollback exception was thrown, which was displayed on the user interface. The correction was made to the CheckDuplicatesWorksheet.pcf file in Studio. The path is config/web/pcf/claim/financials/checks/CheckDuplicatesWorksheet.pcf.
CC-47242	Added a verification that a supplement payment must be marked as non eroding before being saved to the database. This is on the field DoesNotErodeReserves on the Payment entity.
CC-47466	Fixed an issue where an exception was thrown if a bundle rollback failed in the Check wizard.
CC-47917	Fixed an issue when entering an exchange rate, the number of significant digits is now correctly validated.
CC-48020	Improvements were made to the Javadoc in the IClaimFinancialsAPI.gs file.
CC-48176	Corrected an issue where the claimant view of the Financials Summary page having Claim Level reserve lines listed under each claimant resulted in an inconsistent Claim Total on the Claimant view compared with the exposure view. The fix now has Claim Level data not listed in the Claimant view.
CC-48237	Fixed an issue where the <i>FinancialsCalculations</i> batch process calculated the gross amount in the cc_checkrpt table incorrectly for checks that are recoded and then stopped.
CC-48467	Fixed an issue where attaching a document to a new Recovery Reserves set would throw a stacktrace, if no reserves had been edited.
CC-48702	Fixed an issue where approving a Manual Check with a final payment would throw a CloseValidationException error if the claim could not be closed. Now this exception is caught and a warning Activity created, as is done during check escalation.
CC-48794	Fixed an issue in release 5.0.3 where calling the updateCheckStatus method on a <i>PendingVoid</i> or <i>PendingStop</i> check to transition it to <i>Voided</i> or <i>Stopped</i> status incorrectly modified the status of the offset payments. This issue is also fixed in version 5.0.3 patch 2. If you were live on versions 5.0.3 or 5.0.3p1, then you should consult with Guidewire Support to see if this issue affects you.
CC-49044	Fixed an issue where the sum on the Set Reserves page was not correctly calculated on the Change column.
CC-49046	Fixed an issue where the method IClaimFinancialsAPI.updateCheckStatus() now verifies that the current check status is at least a committed status, before applying the new status. This prevents mistaken updates to <i>PendingApproval</i> or <i>AwaitingSubmission</i> checks. However, this could be even more restrictive, so some invalid status updates will still not be prevented. Refer to the ClaimCenter Application Guide for valid check status transitions.
Geocoding	
CC-48111	Corrected an issue where ordinal proximity searches could find incorrect locations. The fix included checking that all addresses found by the initial HTMID filter ensuring they were actually within the believed distance. In the rare cases this filtering error is detected, the ordinal search continues to expand the search radius, checking each set of results.
ListViews	

ID	Description
CC-45863	Fixed an issue where adding a new Vehicle Assessment Item on a Vehicle Incident, while the list of assessments was already being edited, would not update the list with the newly added item. Added a picker expression to the Edit buttons of the new Assessment popup, so the LV now refreshes. Affected files: <ul style="list-style-type: none"> config/web/pcf/claim/assessment/AssessmentItem/NewVehAssessItemPopup.pcf config/web/pcf/claim/assessment/AssessmentItem/VehAssessItemLV.pcf
Litigation Management	
CC-48147	Fixed an issue where a litigation history event was created when the Claim's litigation status was set to <i>Not Litigated</i> .
Metro Police Reports	
CC-46168	When sending an auto accident report to Metro, the request now contains the vehicle information (make, model, and so on) if it is available. In Studio, the affected file is: config/web/templates/metreport/MetroAutoAccidentReport.gs.
New Claim Wizard	
CC-45967	Fixed a potential issue where the claim snapshot could have been bloated with extra data (such as full copies of all the question sets used during the New Claim wizard). Now the snapshots are stored with minimal data.
Performance	
CC-49812	Added the following configuration parameters to control whether the Team Group Activities query uses certain Oracle optimizer hints: <ul style="list-style-type: none"> DisableIndexFastFullScanForTeamGroupActivities DisableCBQTFForTeamGroupActivities DisableHashJoinForTeamGroupActivities DisableSortMergeJoinForTeamGroupActivities
CC-49969	To improve the performance of check searches using certain date values, Guidewire has added indexes checku7 and checku8. These indexes can be modified or removed based on the check searches used in your deployment. The indices are located in the extensions.xml file. Ensure that the performance is not adversely affected before making any changes.
Policy	
CC-30654	Fixed an error where a Null Pointer Exception was thrown when viewing a newly created claim on a policy with 200+ vehicles.
Question Sets	
CC-50011	Corrected the definition in the Data Dictionary for the QuestionFilter entity.
QuickJump Box	
CC-48436	The <i>NewDocumentTemplate</i> QuickJump option is no longer available if you do not have a role that has the Document Create permission.
Services	
PL-319	Corrected an issue that caused incorrect Japanese RTF output if creating output from templates created by the Japanese version of Microsoft Word.
PL-323	Fixed an issue in which multiple attempts to send the same email with a missing subject caused the application server to go into an infinite loop. Affected config file: config/web/pcf/claim/email/CreateEmailScreen.
PL-327	Corrected an issue that did not correctly refresh the Cache Info page.
SOAP APIs	
CC-49377	To change status of a check from messaging plugin code, call the updateCheckStatus domain method on the check. Do not call the SOAP API method updateCheckStatus in the IClaimFinancialsAPI interface. For example, you can use code such as ((Check) Message.getMessageRoot()).updateCheckStatus(null, null, TransactionStatus.VOIDED). WARNING: This release's <i>Integration Guide</i> is incorrect on this subject. Do not call setStatus as the <i>Integration Guide</i> mentions. Instead, call the updateCheckStatus method on the check.

ID	Description
PL-316	Guidewire recommends that you regenerate and compile your web service toolkit JAR files. Guidewire has made several small changes to the Guidewire published server WSDLs in order to address potential external compatibility issues. In particular, in the schema of types on the Guidewire published web services, Guidewire has enacted a consistent policy for ordering type properties in order to ensure interoperability. (Guidewire, in the past, did not enforce the ordered serialization of schema sub-elements and changes.) These changes, most likely, will not affect existing integrations. However, Guidewire does advise that you regenerate and recompile your web services toolkit JAR files.
Studio	
PL-271	This fix changes HTML layout and CSS to improve the response time. For both versions 6 and 7 of Microsoft Internet Explorer, Guidewire recommends that you open all the application screens with unique layouts and look for unexpected changes. This fix affects the spacing and wrapping on the screen. There will likely be more wrapping of labels and text but with less horizontal scrolling. For details refer to: Wrapping of Labels.
PL-272	The Title Bar element has moved to the Panel Layout group in the Page Configuration editor.
PL-273	Corrected an issue with the Studio debugger in which you could not reopen the Debugger Pane (at the bottom of the screen) after you had closed it.
PL-280	Fixed an issue that prevented Studio from opening the ActivityDetailPrint PCF file.
PL-284	Removed the Print Toolbar button from the Studio PCF editor.
PL-288	Fixed an interface issue that caused the Quick Jump menu to render upwards, blocking visibility to the remainder of the items in the QuickJump list. This occurred if the browser vertical space was too small.
PL-296	Fixed an issue that caused Studio to incorrectly handle comments embedded in PCF files.
PL-4102	Fixed an issue in which Studio did not recognize the existence of newly created packages.
Web	
PL-368	Fixed an inconsistency in how the application displays negative money amounts. The Recovery and Payment details screens now show negative money amounts in red, the standard.
PL-369	Corrected an issue with a missing GIF file that caused 404 errors.
PL-371	Fixed a problem adding up the summary row total for a numerical list view column, which could result in the display of too many fractional digits.
PL-373	Fixed an issue in which the method <code>CurrentLocation.startEditing()</code> did not work correctly in a popup window that had a <code>startEditing</code> attribute defined.
PL-4098	Added the ability to navigate into the Contact Picker (from the FNOL wizard) by tabbing.
PL-4227	Corrected an issue in which the application did not display the hourglass icon while the application was busy.

Known Issues and Limitations

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

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER/Software/Microsoft/Windows/CurrentVersion/Explorer/BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin configuration in ClaimCenter/config/config.xml.

Using a message handling plugin to deny a check containing a final payment causes an error (CC-27165)

Issue: Using a message handling plugin to deny a check containing one or more 'final' payments causes an error. Denying checks that contain only 'partial' and 'supplemental' payments via a message handling plugin works correctly.

Workaround: Use IClaimFinancialsAPI to deny a check with a final payment instead of a message handling plugin. Guidewire may fix this issue in a future release.

Very fast execution of the Geocoding writer can result in duplicate work items in the queue (CC-29370)

Issue: Using a cycle time for the Geocoding writer that is too short can result in duplicate work items being created, which will have a negative impact on system performance.

Workaround: The geocoding writer time interval (set in `scheduler-config.xml` for either ContactCenter or ClaimCenter) should never be set to shorter than either:

- The expected time for a batch of new addresses to be successfully geocoded, or
- The expiration time of a geocoding work item check-out (the `progressInterval` attribute in the `<workqueue>` element).

The first case is of particular relevance to upgrades of ContactCenter and to customers newly deploying Geocoding on an existing ContactCenter server; the default batch process time (hourly) will be too short for customers Geocoding a very large number of addresses. Two options are to initiate the first run of the Geocoding writer manually using **System Tools**, or to temporarily use a much longer cycle time than hourly until existing addresses are completely processed.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` method in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a Pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaimAPI depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: `IClaimAPI` depends on `SynchStateData`, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

An exception is thrown when trying to link a new contact to the address book from the new check wizard (CC-32732)

Issue: If you create a new contact in the New Check Wizard, and later attempt to link contacts to ContactCenter, you get an error.

Workaround: None.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To retrieve the payment's send date, both the Transactions and the Checks table have to be searched; this hampers performance and is not done.

Workaround: None.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The **Reset** button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the **Reset** button an **Initialize** button.

After logout, a request to return to the login page creates a new session in Websphere (CC-36394)

Issue: On logout in WebSphere, the current session is destroyed but a new one is created. In ClaimCenter, logging out returns you to the login page. Whenever a browser interacts with an application server, the server creates a session if none existed.

Workaround: On logout, go outside of the Guidewire application and enforce static content caching. The browser will reuse static objects in its cache (like the login page) and therefore minimize the unnecessary interactions with the application server. Alternately, have a web tier cache the static content. This is basically adding another caching layer (on top of the browser) and would reduce this phenomenon further.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you access the ClaimCenter **Desktop Activities** page and repeatedly click **Activities**, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to Internet Explorer 7.

regen-toolkit results in a "code too large" error (CC-37415)

Issue: When you run `regen-toolkit`, you receive the error: `code too large` when it is processing a typelist. This occurs when you have a typelist that is too large.

Workaround: Reduce the size of the typelist. The typelist size depends on the number of typekeys and the length of the strings (code, name, description) used to define them. You can reduce the typelist size by reducing any of these (for example, you can include more typekeys by shortening the description text).

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the **Plan of Action** → **New Evaluation** page will display an incorrect title *New Evaluation* instead of *LOBName Evaluation*, and the page will not have a *LOBName Cost* field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use `ClaimEvaluationDetail` page instead of `ClaimEvaluationDetails<LOB>` page. Also, do not customize the `ClaimEvaluationDetail` page heavily.

Acrobat Sample document template does not allow extra fields (CC-40462)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

After upgrading from ClaimCenter 3.1, the doctor is no longer shown in medical detail exposures (CC-40595)

Issue: ClaimCenter 3.1 displayed the value `Exposure.Doctor`, but later versions of ClaimCenter display the value `Claim.FirstIntakeDoctor`. During upgrade, the data is not migrated from the old field to the new field, resulting in the upgraded ClaimCenter version showing a blank instead of the real value.

Workaround: Add another PCF field in ClaimCenter 5.0 on the **Medical Details** page below the current **Doctor** field for `Exposure.Claim.FirstIntakeDoctor`. This displays the upgraded data.

ClaimCenter uses an old version of Castor libraries (CC-40659)

Issue: Guidewire products still use an old version of the Castor library, and you cannot modify the version used. The version in ClaimCenter is bundled version 0.9.4.3. The current version of the Castor libraries is 1.2.

Workaround: You can use another version of this library within your plugin code. To use a different version of Castor, add its jar to the `modules/configuration/plugins/<plugindir>/lib` folder. All jar files located in the plugins directory are loaded with a different class loader than the one used by the ClaimCenter core code. This avoids any potential conflicts between versions of the libraries.

Potential to lose LocationXXX data during upgrade (CC-41555)

Issue: Guidewire changed the location of the `LocationCity`, `LocationState`, `LocationStreet`, and `LocationZip` fields between ClaimCenter 3.1.x, 4.0.x, 5.0.0, and 5.0.1:

- In ClaimCenter 3.1.x, these fields live on the exposure.
- In ClaimCenter 4.0.x, these fields live as extensions on `MobilePropertyIncident`.
- In ClaimCenter 5.0.0, Guidewire moved these fields into a separate `LocationAddress` object, in the core data model.
- In ClaimCenter 5.0.1, Guidewire changed `LocationAddress` into an extension on `MobilePropertyIncident` (in `extensions.xml`).

Upgrading from ClaimCenter 3.1.x or 4.0.x. Guidewire added a version check that should prevent data loss when upgrading to 5.0.2 from either 4.0.x or 3.1.x. It will look for the `LocationXXX` fields in incident or exposure (depending on whether you are upgrading from 4.0.x or 3.1.x). If any of those fields exist, then the version check will insist the following:

1. There is a `LocationAddress` field, of type `Address`, somewhere in the incident hierarchy
2. That any incidents/exposures with the `LocationXXX` fields set map to incident types that contain the `LocationAddress` field. That is, it checks that all `LocationXXX` fields that contain a value have a `LocationAddress` field to use in migrating the data.

If either of these checks fail, the version check will abort the upgrade and you must then follow the steps outlined in the workaround to solve the problem.

Upgrading from ClaimCenter 5.0.0. If you are upgrading from ClaimCenter 5.0.0, there is already a `LocationAddress` field. However, in 5.0.0, Guidewire placed the field in the core data model, while in 5.0.1 and later it is in `extensions.xml`. So, if your target version `extensions.xml` does not contain a `LocationAddress` field, then the `LocationAddress` field could get dropped entirely during the upgrade process, possibly losing data. This is the case that the upgrade version trigger does not address.

Upgrading from ClaimCenter 5.0.1 or later. This is not an issue.

Workaround: Depending on the release from which you are upgrading, you need to do the following:

- **ClaimCenter 3.1.x base release.** In 3.1.x, the `LocationXXX` fields were in the base data model, on exposure. As part of the upgrade, you will need to decide which exposure types map to which incident subtype. You must map any exposure types that have the `LocationXXX` fields set to an incident subtype that has the `LocationAddress` field. This ensures that the upgrade process has a place to move the data.

- **ClaimCenter 4.0.x base release.** In 4.0.x, the LocationXXX fields were in extensions.xml, in the incident hierarchy. You could choose where in the incident hierarchy to put these fields. In upgrading to any 5.0.x version, replace these fields in extensions.xml with LocationAddress, at the same level in the hierarchy.
- **ClaimCenter 5.0.0 base release.** In 5.0.0, the LocationAddress field was in the base data model, in MobilePropertyIncident. If you are upgrading to 5.0.x, you will need to make sure LocationAddress is in the extensions.xml file. Guidewire recommends that you put it in MobilePropertyIncident. However, if desired, it is possible—because it now lives in extensions.xml—to move it to a different place in the incident hierarchy.

Transaction runWithNewBundle Does Not Follow Links (CC-42654)

Issue: The GScript API that runs a block of code with new bundle (`gw.transaction.Transaction.runWithNewBundle(...)`) has some limitations. Even if you call `newbundle.add(yourentity)`, the add method does not follow the graph links to other entities to which it links. You must manually add each sub-object to the new bundle, using the call `bundle.add(linkedEntity)` for each entity. This includes all foreign key references to other entities as defined in the built-in data model files, but also includes customer data model extensions that include foreign key references.

Workaround: You can work around this by committing all entities in the current bundle instead of using this API. This may be an issue when designing implementing a custom web service. Committing all entities in the current bundle includes all entities serialized into the bundle as part of the web service request because they were parameters to a web service method. There may be cases where entities that are web service API parameters you do **not** want commit to the database.

One workaround is to redesign the web service API so instead of taking entities (the entities you do not want to commit) it takes GScript class instances instead of Guidewire entities. Your custom GScript classes can provide only the fields that you want to pass in your integration code, which may be a much smaller subset of fields in some cases. Because they are GScript classes and not entities, committing the entire bundle with `gw.transaction.Transaction.getCurrent().commit()` will skip that data that you choose not to commit to the database.

Creating a new document from a template results in a ClassNotFoundException (CC-42867)

Issue: If you implement the IDocumentProduction plugin in Java, then when you create a new document from a template you may encounter a ClassNotFoundException.

Workaround: Copy the file `ClaimCenter/modules/cc/config/plugin/ExternalMappings.properties` to `ClaimCenter/modules/configuration/config/plugin`. Then add the following line to the copied file:

```
gw.api.document.DocumentCreationInfo = gw.api.document.DocumentCreationInfo
```

In addition, copy the file `ClaimCenter/modules/cc/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf` to `ClaimCenter/modules/configuration/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf`. In the copied file, replace the following line:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.DocumentTemplateDescriptor
      .TemplateId, gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

with the following:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    util.document.DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.
      DocumentTemplateDescriptor.TemplateId,
      gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```


After upgrade, two duplicate incidents are created (CC-44646)

Issue: If you have two exposures with the same incident type details (for example, auto damage on the same car with the same driver), the database upgrader creates two identical incidents.

Workaround: Please contact Guidewire Support for assistance.

Type name conflict in web services (CC-45132)

Issue: It is illegal to have two types with the same relative name within the context of the same web service, or within the context of any published web service.

Workaround: Use unique names.

Limitation in reporting when a typelist name contains a comma (CC-45305)

Issue: Due to the limitations of the InetSoft reporting software, issues arise when a passing a typelist as a parameter to a drill-down report if one of its typecode values contains a comma. (For example, typelist Loss Cause contains typecode *Fall, slip, or trip injury*.) If a typecode does contain a comma, InetSoft thinks it is an array and sends multiple values. Guidewire has implemented a solution for the Loss Cause typelist in ClaimCenter. However, if you are passing any other typelist as a parameter to a drill-down report and it contains a typecode with a comma, you will experience this issue.

Workaround: Guidewire is aware of this issue and will address it in a future release. If you encounter this issue with any other report parameters, contact Guidewire Support for assistance.

Cannot upgrade to ClaimCenter 5.0.3 if you implemented archiving in an earlier release (CC-45993)

Issue: If you are using the archiving feature in ClaimCenter 5.0.0 through 5.0.2, you cannot upgrade to ClaimCenter 5.0.3.

Workaround: Contact Guidewire Support for assistance. If you start using archiving in ClaimCenter 5.0.3, there will be no issues with future upgrades.

Typelist error message: "contains a typecode with an empty code/name" (CC-46085)

Issue: When starting the server, you may see an error that says a typelist contains a typecode with an empty code or name. This error is mistakenly generated and does not indicate any problem.

Workaround: Ignore this error.

Upgrade fails when upgrading CoverageType typelist that already has a code of: BLDG (CC-46510)

Issue: During upgrade, you receive an exception for the following upgrade trigger: `RecodeCoverageTypecodesTriggers$RecodeCoverageTypecode` and your Coverage Type typelist already has a typekey with code =BLDG

Workaround: Contact Guidewire Support for assistance.

Deny Check associated with a bulk invoice item is not supported (CC-46653)

Issue: Denying a check associated with a Bulk Invoice item (`check.Bulked == true`) results in an error. The Deny Check feature does not support such checks.

Workaround: Such checks share the same characteristics as the Bulk Invoice or Bulk Invoice item, so in the Bulk Invoice Validation plugin, check for the conditions that could lead to Denial downstream.

Unable to subclass `gw.api.quickjump.SpecificClaimCommand` (CC-49433)

Issue: If you extend the `gw.api.quickjump.SpecificClaimCommand` class to override the `isPermitted()` method, then the Specific Claim QuickJump command is unavailable. It also can cause an error during server startup.

Workaround: You can create a new QuickJump command that subclasses `com.guidewire.pl.web.navigator.commands.DefaultQuickjumpCommand`, and which delegates its operations to an instance of the `gw.api.quickjump.SpecificClaimCommand`. If you need to create a `SpecificClaimCommand`, contact Guidewire Support for sample code attached to issue PL-4061.

Iterator buttons not working correctly in `MatterDetailsDV.pcf` (CC-48879)

Issue: On the `MatterDetailsDV` file, the **Add** button does not hide when not in *Edit* mode. Also, the **Add** and **Remove** buttons do not show when in *Edit* mode, on the `StatusLinesLV` file.

Workaround: Set `lockWhileEditing="false"`. Note it is not locked.

`ShowNewExposureChooseByCoverageMenuForLossTypes` parameter does not allow a blank value (CC-49426)

Issue: If the `ShowNewExposureChooseByCoverageMenuForLossTypes` configuration parameter is not defined in the `config.xml` file, then the server throws an exception trying to look for at least one loss type in this parameter. This is problematic if you do not want the menu to be displayed.

Workaround: Remove the menu from the PCF files.

FNOL wizard not saving selected vehicle (CC-47439)

Issue: The second step of the FNOL wizard does not save the selected vehicle when viewing the **Policy Details** screen and returning to the **Basic Info** screen.

Workaround: In the `FNOLWizard_BasicInfoScreen.default.pcf` file, there is an `InputGroup` with ID `InsuredVehicleInputGroup`. Add an `onToggle` attribute.

Integrity checks allow offsetting payments to be associated to a different check than the payment it offsets (CC-49866)

Issue: Integrity checks for staging tables do not ensure that the two payments pointed to by a `TransactionOffsetOnset` row belong to the same check.

Workaround: You must correct the errors in data mapping, and ensure that each Voided or Stopped payment on a Voided or Stopped check also has an offset payment (with negative amount, in Submitted status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

Exception stack trace in user interface when `DisplayableException` thrown from Policy plugin (CC-47987)

Issue: A GScript util class throws a `DisplayableException` to the policy plugin GScript class which is again throwing another `DisplayableException` with the message from the first exception. The exception message is not user friendly.

Workaround: Add the following parameter to your `config.xml` file:

```
<param name="IntegratedStackTraces" value="true"/>
```


ArithmeticException when exchange rate is set to 0 (CC-48438)

Issue: When you enter an exchange rate = 0 in the New Check wizard, you see an `ArithmeticException` error in the user interface.

Workaround: Add a validation expression (attribute `validationExpression` or `requestValidationExpression`) to the `Transaction_ExchangeRate` element in `ExchangeRateInputSet.default.pcf` and `ExchangeRateInputSet.Check.pcf` files.

Sessions which have idled and been reaped still appear as active (PL-1699)

Issue: When a session times out, it still appears as active. This can be seen in the Management Bean for viewing active user sessions in the system.

Workaround: None. Guidewire is aware of the issue and will address it in a future release.

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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 info, 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 `checkForDuplications` in the `NewClaimWizard.pcf` file as seen in the following example:

```
function checkForDuplications() {
    if (Wizard.checkForNewDuplicateClaims()) { NewClaimDuplicationsWorksheet.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="checkForDuplications()"
    ...
```

To allow **Next** to execute, you can create a new function such as:

```
function checkForDuplicationsNoWarning() {
    if (Wizard.checkForNewDuplicateClaims()) {
        NewClaimDuplicationsWorksheet.goInWorkspace(Wizard)
    }
}
```

Have the `onExit()` function of the Main Contacts wizard step call:

```
onExit="checkForDuplicationsNoWarning()"
```

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.

Question Set Filters do not function correctly (PL-4658)

Issue: Question set filters do not filter correctly in ClaimCenter.

Workaround: Perform the following:

1. In Studio, copy the GScript files (that are attached to this issue) under **Resources** → **Classes** → **libraries**. In the `QuestionSetLV.pcf` file, set the `visible` property of the list view to:
`QuestionSet.isQuestionSetAvailable_Ext(AnswerSetContainer)`
2. In the .pcf file that has the question sets included, replace `QuestionSet.isQuestionSetAvailable(Claim)` with `QuestionSet.isQuestionSetAvailable_Ext(Claim)`.

Error when removing claim flags through the Team tab (CC-50159)

Issue: The code to remove the flagged status from a claim in the Team view only works with one claim at a time. The following PCFs allow you to select multiple claims in the LV and then click **Remove Flag**, which results in an error. This can happen with these PCFs: TeamGroupOtherClaims.pcf, TeamGroupPendingClaims.pcf, TeamUserClaims.pcf

Workaround: Modify the `flags` attribute of the Remove Flag button in each PCF to read:

```
flags="one CanRemoveFlag"
```

chapter 29

Guidewire ClaimCenter 5.0.5 Release Notes

Release 5.0.5.14

Copyright © 2001-2009 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

- This release of Guidewire ClaimCenter is 5.0.5.14
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 9.0, build 073008.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter

The following issues pertain to installing this release.

Inetsoft Evaluation

You can evaluate Inetsoft Style Report Enterprise Edition Version 9.0 using the following license key:

L000-69C-ERX-000097000006-F2A7343ACB01

If you want to use Inetsoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide*.

Software Updates

There are several software updates:

- This release supports Microsoft Internet Explorer 8
- If you use SQL Server 2005, you must install its Service Pack 3 (SP3)
- You must run 64-bit operating systems on production application servers

Archiving Updates

Beginning in this release, Guidewire has included additional checks to be run on the claim graph. You can view the results of these checks by accessing **Internal Tools**. While you might not go live with archiving in this release, Guidewire recommends that you resolve these warnings before going live on ClaimCenter 5.0.5.

See the following issues for additional details:

- PL-6082
- CC-51977

Also, there are other issues in this release that pertain to archiving claims:

- PL-6311
- CC-51808
- CC-45993

Changes to HTML Tables

Internet Explorer's rendering of HTML tables may cause labels to wrap more aggressively than in previous ClaimCenter versions. The wrapping occurs most commonly in the left-most columns of detail views, but can appear in other places throughout the product interface. The issue was explained in the previous (ClaimCenter 5.0.4) release notes under the section *Wrapping of Labels*.

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Configuration Parameter Changes
- Improvements and General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 5.0.4 and ClaimCenter 5.0.5, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- GScript API
- typelists
- display keys

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.4 to 5.0.5

- 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/platform` directory, [click here](#).

Base Resource Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.4 to 5.0.5

- To view a report of the changes in the base resources in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base resources in the `modules/core` directory, [click here](#).
- To view a report of the changes in the base resources in the `modules/platform` directory, [click here](#).

Configuration Parameter Changes

The following are the primary changes to the ClaimCenter configuration parameter default settings specified in `config.xml`:

- Added the following new parameters:
 - `BucketSizeForHistogramsOnAllIndexedColumns` — If `CollectHistogramsOnAllIndexedColumns` is true, then set the bucket size for histograms.
 - `ClusterProtocolStackOption1` — This is a local option that can contain other parameters for the `ClusterProtocolStack`.
 - `ClusterProtocolStackOption2` — This is a local option that can contain other parameters for the `ClusterProtocolStack`.
 - `CollectHistogramsOnAllIndexedColumns` — Collect histograms on all indexed columns instead of specific columns. Applicable only to Oracle.
 - `DefaultEvictTimeMinutes` — The default evict time of objects in the cache.

Improvements and General Issues

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

ID	Description
Aggregate Limits	
CC-50368	Fixed a potential performance issue that could occur when adding policy periods to a policy that was linked to a large number of claims.
Application Framework	
PL-425	Corrected several issues with the <code>IDocumentMetdataSource</code> plugin. This included an issue in which creating multiple documents (without navigating away from the initial page) could potentially create two documents with the same identical name. Also, if attempting to change a document from <i>Draft</i> to <i>Final</i> , the document could still show the <i>Draft</i> status with the Edit button visible in the document screen.
PL-428	Attempting to use the Guidewire Create Document feature on the Microsoft Vista platform opens a Security Warning dialog. To correct this problem, you must add the Guidewire application server to the client computer's <i>Trusted Sites</i> security zone.
PL-4231	Corrected an issue in which the server, attempting to access a non-existent document template, generated an exception. The current behavior now generates an error message and logs the name of the missing form template.
PL-4426	Modified the behavior of method <code>assignUserByLocationUsingProximityAndAttributes(address, criteria, includeSubGroups, currentGroup)</code> to return <code>null</code> if either one of the following conditions occurs: <ul style="list-style-type: none"> • Both <code>address</code> and <code>criteria</code> are <code>null</code>. • All four arguments are <code>null</code>.
PL-4523	Updated the PDF Library to correct a security issue for PDFs containing JavaScript.
PL-4888	Previously, the <code>Assignable.assignUserByProximityWithSearchCriteria</code> method would determine the set of assignable users without regard to whether a user was inactive or on vacation. This could cause problems if the set of potential users only included these kinds of inactive users and would cause the assignment to fail. The assignment method now restricts the user query to those users that are active.
PL-4945	Guidewire added the following new methods to the <code>IAddressBookAdapter</code> API. For details, see the JavaDoc. <ul style="list-style-type: none"> • <code>retrieveRelatedContacts(contact, contactRelationshipSpec)</code>, which retrieves related contacts for a given contact. • <code>loadContact(contact, onlyPrimaryRelationships)</code>, which loads the given partially populated contact into the current edit context, with a flag to indicate whether to load all related contacts or just primary relationships. • <code>populateRelatedContacts(Contact contact)</code>, which loads the remaining related contacts into the provided contact, which must be in the current edit context.
PL-4975	Modified <code>assignGroupByRoundRobin</code> so that it now ignores any group for which the <code>Group.LoadFactor</code> value is zero (0).
PL-4982	Modified the <code>assignGroupByRoundRobin</code> assignment method to perform round-robin assignment with consideration of <code>Group.LoadFactor</code> values regardless of whether the <code>GroupType</code> parameter is <code>null</code> . (Previously, it did not perform round-robin assignment if the <code>GroupType</code> parameter was <code>null</code> .)

ID	Description
PL-5149	Added new dependency.checker.interval property to sree.properties for Standard Reporting. This value sets the dependency check rate to once per day. The default for this value is twice a minute, which can cause performance problems.
Claim Archiving	
CC-51977	Added more validation around the archive graphs. In this 5.0.5 release, a new check was added: validateEntitiesOutsideOfDomainGraphAreNotExtractable. This check, as well as existing ones, do not execute unless archiving is enabled.
Claim File	
CC-50671	Corrected an issue where removing associated claims and re-adding them on the Claim Association Detail page without getting a stack trace. The original public method ClaimAssociation.removeFromClaimsInAssoc() removed the selected claims, and deleted the entire ClaimAssociation if there were no associated claims. Guidewire updated the public method to only remove the selected claims. The logic for deleting the entire ClaimAssociation if no associated claims existed has been moved to only occur during a claim purge.
CC-51272	Corrected an issue that allowed you to use staging table loading to create a loss location that is shared between claims. If you use GScript to add a loss location to a claim that is already used, a consistency check warns you with the message: A single loss location is being shared by more than one claim.
Core	
PL-2202	Corrected an issue in which importing administration (user) data would incorrectly overwrite existing data. This occurred even if you specified <i>do not</i> overwrite data during a merge resolution.
PL-4414	Guidewire has added two new configuration parameters that you can use to set a specific bind address on a local node. They are: <ul style="list-style-type: none"> ClusterProtocolStackOption1 ClusterProtocolStackOption2
PL-4533	Guidewire modified how it handles objects in the data object cache. Now, objects have an evict time that determine the point at which the application removes an object from the cache. This is different from the currently existing stale time value that the application uses to determine if it needs to retrieve a fresh version of an object from the database. To manage the eviction time, Guidewire added a new configuration parameter (DefaultEvictTimeMinutes). See the section on <i>Cache Management</i> in the System Administration Guide for more information.
PL-5641	Guidewire added 100 Profiler tags for use by customers to the base application configuration. This includes: <pre>public static final ProfilerTag CUSTOMER_00 = new ProfilerTag("Customer_00", "Customer Tag _00"); ... public static final ProfilerTag CUSTOMER_99 = new ProfilerTag("Customer_99", "Customer Tag _99");</pre>
PL-6082	There can be warnings related to the Claim graph, which can occur for many different reasons, including data model changes, business logic, or other reasons. As there is no single cause for these warnings, there is no single solution or workaround. You can see these warning messages in the Internal Tools user interface. Navigate to the Warnings tab on the Archiving Info screen. Guidewire recommends that you investigate and resolve each one individually as necessary.
PL-6311	Corrected an issue in which archiving a Contact along with the Claim that referenced it could cause a foreign key constraint violation, especially if ClaimCenter was integrated with ContactCenter. This could potentially cause the archive process to fail.
Database	
PL-432	You must install Service Pack 3 (SP3) for SQL Server 2005 database if you use SQL Server 2005.
PL-1554	Guidewire includes a newer version of the Microsoft SQL Server JDBC driver with this release. This driver corrects a performance issue under heavy concurrent use. If you use an external connection pool, copy this new JDBC driver, webapps/cc/WEB-INF/lib/sqljdbc.jar, to the location where the connection pool obtains its driver. For WebLogic, this location is typically the server/lib directory. For WebSphere, this location is typically the lib/ext directory.
PL-4750	Fixed a problem in which a minor upgrade that did not change any physical database tables, but only meta-data such as adding an array to an extension entity, would leave the database in an inconsistent state.

ID	Description
PL-5490	<p>There are two new configuration parameters to control Oracle database statistics gathered using <code>maintenance_tools</code> options. They enable the creation of histograms on all indexed columns with the optional bucket size. The configuration parameters are:</p> <ul style="list-style-type: none"> • <code>CollectHistogramsOnAllIndexedColumns</code> • <code>BucketSizeForHistogramsOnAllIndexedColumns</code> <p>The default of <code>CollectHistogramsOnAllIndexedColumns</code> is set to <i>false</i> as the preferred method is to gather histograms on select columns.</p>
Documentation	
PL-4318	<p>Previous documentation misstated the default scope for static variables in GScript. The default scope for static variables is <i>application</i> scope, not <i>session</i> scope. Avoid static variables if you can to avoid issues with multiple thread access to the same static data. If you cannot avoid static variables, synchronize access to structures stored in static variables so your code is thread-safe. Refer to the <i>Plugins</i> topic in the Integration Guide for more information about synchronization.</p>
PL-5237	<p>Guidewire has moved postal code field validation from the <code>fieldvalidators.xml</code> file to the <code>address-config.xml</code> file.</p>
PL-5356	<p>Guidewire imposes an upper limit of 8000 typecodes on a typelist. If you attempt to create a typelist with a larger number of typecodes, then the application refuses to compile.</p>
Financials	
CC-47338	<p>In ClaimCenter 5.0.5, Guidewire deprecated the Java interface <code>gw.api.webservice.cc.financials.IClaimFinancialsAPI</code>. It was a Java interface used to publish the <code>IClaimFinancialsAPI</code> SOAP API in ClaimCenter 4.0.X versions. ClaimCenter no longer uses this Java interface to publish the SOAP API. If you upgrade to ClaimCenter 5.0.5, you do not need to take any upgrade action in this release.</p>
CC-48661	<p>Corrected an issue where the deduction amount was not updated after a secondary payee check was either removed or added.</p>
CC-49027	<p>Corrected an issue so that you could update the status of a bulk invoice to <i>voided</i> from within the <code>send</code> method of a messaging transport. The fix included adding a new method on <code>BulkInvoice</code>: <code>updateBulkInvoiceStatus</code>. Guidewire strongly recommends using this from messaging plugins to update the status of the bulk invoice in only this specific context: From messaging plugins after calling <code>message.reportAck()</code> to acknowledge the message. This would be either in your <code>MessageTransport</code> plugin after a synchronous send and acknowledgement or in your <code>MessageReply</code> plugin after an asynchronous acknowledgement of the message.</p> <p>Guidewire strongly admonishes to not use the methods <code>check.updateCheckStatus()</code> or <code>bulkinvoice.updateBulkInvoiceStatus()</code> from any other context.</p>
CC-50543	<p>Corrected an issue where two different users modifying the same transaction simultaneously could affect the same T-accounts as in two users trying to approve a pending-approval transaction at the same time. The fix included notifying the second user with an error message instructing the user to retry the update. You can change the error message which is located in the <code>Financials.Transaction.Error.ConcurrentDataChange</code> display key.</p> <p>Affected file: <code>config/locale/en_US/display.properties</code></p>
CC-51281	<p>Fixed an issue of being unable to create zero amount reserves in rules through the public methods:</p> <ul style="list-style-type: none"> • <code>Exposure.createInitialReserves</code> • <code>Claim/Exposure/ReserveLine.setAvailableReserves</code> • <code>Exposure/ReserveLine.setOpenRecoveryReserves</code> <p>Guidewire now allows a zero amount reserve and recovery reserve to be created, but only if there are no existing transactions with that reserveline.</p>
CC-52222	<p>Fixed an issue where canceling out of the first editing session that used the current market Exchange Rates could kick them out of their bundle, resulting in stack traces when trying to use them until the exchange rates cache was refreshed.</p>
GScript	
PL-4074	<p>Corrected an issue in which using a virtual property in a GScript find expression looked valid in Studio but failed at run-time. The GScript editor now indicates such code as invalid (the validation icon is red), and displays a red line in the margin next to the invalid line of code. A tooltip displays an error message.</p>
PL-4302	<p>There is a new query system that improves upon the feature set of <i>find</i> operations. It has a syntax more similar to SQL and supports advanced joins and subselects. See the "Query Builder" topic in the GScript Reference Guide for more information.</p>

ID	Description
Internationalization	
CC-52157	Corrected an issue that occurred when the display names for <code>Java.Financials.Summary.Create.Check</code> , <code>Java.Financials.Summary.Create.QuickCheck</code> , and <code>Java.Financials.Summary.Create.Reserve</code> had UTF-8 characters. This caused an issue in the Financials Summary screen when you clicked Edit Reserve and you would go to the Quick Check screen instead of the Reserves screen.
ISO	
CC-50239	An improvement was made to the default configuration payload generation for first or third party payload generation. The <code>ISO.gs</code> file now contains methods that improves ISO support for third-party property exposures. The new version relies on the <code>Exposure.LossParty</code> property to be set correctly. This property contains one of two typecodes that indicate <code>Insured</code> (first party loss) or <code>ThirdParty</code> (third party loss). If you modified <code>ISO.gs</code> , Guidewire recommends merging in this change to get the new behavior. The built-in application code maintains the <code>exposure.LossParty</code> property. If any of your customizations affect the status of this property, verify your code sets it property so that the ISO payload generation rules can correctly read that value.
Performance	
CC-51739	To enhance performance on the Team Group screen, an index was moved from the <code>dm_cc_activity.xml</code> file to the <code>extensions.xml</code> file. Also, two additional columns (<code>ClaimID</code> and <code>Priority</code>) were added to the index.
Printing	
CC-51667	Fixed an issue that occurred in the user interface when you returned to a list view that was generated from a picker after clicking Export (as CSV) or Custom Export (as CSV).
Reporting	
PL-5520	<p>Corrected an issue in which Guidewire Standard Reporting displayed the local client currency (meaning the default currency of the operating system) instead of the actual transaction and reporting currency. You now set the default currency symbol for Inetsoft through the <code>sree.properties</code> file. Use the following properties to set the default currency: language and country. In the base configuration, Guidewire sets the default values for these two properties to <code>en</code> and <code>US</code> (English, US), which displays the \$ sign for the currency symbol.</p> <p>To show:</p> <ul style="list-style-type: none"> The British Pound, set these properties to <code>en GB</code>, (English, Great Britain) The Euro, and countries that uses the Euro, use the following values, for example: <code>fr FR</code> (French, France) or <code>de DE</code> (German, Germany). <p>This change is for ClaimCenter 5.0.5 and all future releases.</p>
Statistics	
CC-50954	Corrected an issue where the open matter count, open activity count, open claim count, and open exposure count for a user is incorrect after subsequent calls to the Statistics batch process. The fix also corrected potential synchronization issue when retrieving the item counts as seen on the Team tab.
CC-50976	Corrected an issue where the number of open activities, claims, exposures, and matter for a given group is counted twice when the item is assigned to a group, but not to a user. You can view this from the Team tab.
Studio	
PL-5398	Corrected an issue in which accessing Context Help (CTRL+Q) on a constructor showed the Javadoc for the class, instead of the Javadoc for the constructor. Context Help now works correctly on constructors.
Web	
PL-4517	Corrected an issue such that if the application tried to display a single entity record across multiple rows using a <code>RowIterator</code> , with each column in the record on a separate row, then the rendering engine would create additional, unnecessary blank columns in the first row.
PL-5993	Corrected several Javascript bugs to support Microsoft Internet Explorer 8.

Known Issues and Limitations

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

Note: The maintenance release policy of Guidewire ClaimCenter is to avoid fixing configuration issues that would necessitate merging of files during a maintenance release upgrade. These issues have workarounds that can be employed directly by customers, as seen in this section. The goal with this policy is to make upgrades as straightforward as possible.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER/Software/Microsoft/Windows/CurrentVersion/Explorer/BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Using a message handling plugin to deny a check containing a final payment causes an error (CC-27165)

Issue: Using a message handling plugin to deny a check containing one or more 'final' payments causes an error. Denying checks that contain only 'partial' and 'supplemental' payments via a message handling plugin works correctly.

Workaround: Use `IClaimFinancialsAPI` to deny a check with a final payment instead of a message handling plugin. Guidewire may fix this issue in a future release.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` method in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a Pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaimAPI depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: `IClaimAPI` depends on `SynchStateData`, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

An exception is thrown when trying to link a new contact to the address book from the new check wizard (CC-32732)

Issue: If you create a new contact in the New Check Wizard, and later attempt to link contacts to ContactCenter, you get an error.

Workaround: None.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To retrieve the payment's send date, both the Transactions and the Checks table have to be searched; this hampers performance and is not done.

Workaround: None.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The Reset button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the Reset button an Initialize button.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you access the ClaimCenter Desktop Activities page and repeatedly click Activities, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to Internet Explorer 7.

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the Plan of Action → New Evaluation page will display an incorrect title *New Evaluation* instead of *LOBName Evaluation*, and the page will not have a *LOBName Cost* field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use `ClaimEvaluationDetail` page instead of `ClaimEvaluationDetails<LOB>` page. Also, do not customize the `ClaimEvaluationDetail` page heavily.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

After upgrading from ClaimCenter 3.1, the doctor is no longer shown in medical detail exposures (CC-40595)

Issue: ClaimCenter 3.1 displayed the value `Exposure.Doctor`, but later versions of ClaimCenter display the value `Claim.FirstIntakeDoctor`. During upgrade, the data is not migrated from the old field to the new field, resulting in the upgraded ClaimCenter version showing a blank instead of the real value.

Workaround: Add another PCF field in ClaimCenter 5.0 on the Medical Details page below the current Doctor field for `Exposure.Claim.FirstIntakeDoctor`. This displays the upgraded data.

Potential to lose LocationXXX data during upgrade (CC-41555)

Issue: Guidewire changed the location of the `LocationCity`, `LocationState`, `LocationStreet`, and `LocationZip` fields between ClaimCenter 3.1.x, 4.0.x, 5.0.0, and 5.0.1:

- In ClaimCenter 3.1.x, these fields live on the exposure.
- In ClaimCenter 4.0.x, these fields live as extensions on `MobilePropertyIncident`.

- In ClaimCenter 5.0.0, Guidewire moved these fields into a separate `LocationAddress` object, in the core data model.
- In ClaimCenter 5.0.1, Guidewire changed `LocationAddress` into an extension on `MobilePropertyIncident` (in `extensions.xml`).

Upgrading from ClaimCenter 3.1.x or 4.0.x. Guidewire added a version check that should prevent data loss when upgrading to 5.0.2 from either 4.0.x or 3.1.x. It will look for the `LocationXXX` fields in incident or exposure (depending on whether you are upgrading from 4.0.x or 3.1.x). If any of those fields exist, then the version check will insist the following:

1. There is a `LocationAddress` field, of type `Address`, somewhere in the incident hierarchy
2. That any incidents/exposures with the `LocationXXX` fields set map to incident types that contain the `LocationAddress` field. That is, it checks that all `LocationXXX` fields that contain a value have a `LocationAddress` field to use in migrating the data.

If either of these checks fail, the version check will abort the upgrade and you must then follow the steps outlined in the workaround to solve the problem.

Upgrading from ClaimCenter 5.0.0. If you are upgrading from ClaimCenter 5.0.0, there is already a `LocationAddress` field. However, in 5.0.0, Guidewire placed the field in the core data model, while in 5.0.1 and later it is in `extensions.xml`. So, if your target version `extensions.xml` does not contain a `LocationAddress` field, then the `LocationAddress` field could get dropped entirely during the upgrade process, possibly losing data. This is the case that the upgrade version trigger does not address.

Upgrading from ClaimCenter 5.0.1 or later. This is not an issue.

Workaround: Depending on the release from which you are upgrading, you need to do the following:

- **ClaimCenter 3.1.x base release.** In 3.1.x, the `LocationXXX` fields were in the base data model, on exposure. As part of the upgrade, you will need to decide which exposure types map to which incident subtype. You must map any exposure types that have the `LocationXXX` fields set to an incident subtype that has the `LocationAddress` field. This ensures that the upgrade process has a place to move the data.
- **ClaimCenter 4.0.x base release.** In 4.0.x, the `LocationXXX` fields were in `extensions.xml`, in the incident hierarchy. You could choose where in the incident hierarchy to put these fields. In upgrading to any 5.0.x version, replace these fields in `extensions.xml` with `LocationAddress`, at the same level in the hierarchy.
- **ClaimCenter 5.0.0 base release.** In 5.0.0, the `LocationAddress` field was in the base data model, in `MobilePropertyIncident`. If you are upgrading to 5.0.x, you will need to make sure `LocationAddress` is in the `extensions.xml` file. Guidewire recommends that you put it in `MobilePropertyIncident`. However, if desired, it is possible—because it now lives in `extensions.xml`—to move it to a different place in the incident hierarchy.

Transaction runWithNewBundle Does Not Follow Links (PL-1716)

Issue: The GScript API that runs a block of code with new bundle (`gw.transaction.Transaction.runWithNewBundle(...)`) has some limitations. Even if you call `newbundle.add(yourentity)`, the `add` method does not follow the graph links to other entities to which it links. You must manually add each sub-object to the new bundle, using the call `bundle.add(linkedEntity)` for each entity. This includes all foreign key references to other entities as defined in the built-in data model files, but also includes customer data model extensions that include foreign key references.

Workaround: You can work around this by committing all entities in the current bundle instead of using this API. This may be an issue when designing implementing a custom web service. Committing all entities in the current bundle includes all entities serialized into the bundle as part of the web service request because they were parameters to a web service method. There may be cases where entities that are web service API parameters you do **not** want commit to the database.

One workaround is to redesign the web service API so instead of taking entities (the entities you do not want to commit) it takes GScript class instances instead of Guidewire entities. Your custom GScript classes can provide

only the fields that you want to pass in your integration code, which may be a much smaller subset of fields in some cases. Because they are GScript classes and not entities, committing the entire bundle with `gw.transaction.Transaction.getCurrent().commit()` will skip that data that you choose not to commit to the database.

Creating a new document from a template results in a `ClassNotFoundException` (CC-42867)

Issue: If you implement the `IDocumentProduction` plugin in Java, then when you create a new document from a template you may encounter a `ClassNotFoundException`.

Workaround: Copy the file `ClaimCenter/modules/cc/config/plugin/ExternalMappings.properties` to `ClaimCenter/modules/configuration/config/plugin`. Then add the following line to the copied file:

```
gw.api.document.DocumentCreationInfo = gw.api.document.DocumentCreationInfo
```

In addition, copy the file `ClaimCenter/modules/cc/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf` to `ClaimCenter/modules/configuration/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf`. In the copied file, replace the following line:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.DocumentTemplateDescriptor.
      TemplateId, gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

with the following:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    util.document.DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.
      DocumentTemplateDescriptor.TemplateId,
      gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

After upgrade, two duplicate incidents are created (CC-44646)

Issue: If you have two exposures with the same incident type details (for example, auto damage on the same car with the same driver), the database upgrader creates two identical incidents.

Workaround: Please contact Guidewire Support for assistance.

Type name conflict in web services (CC-45132)

Issue: It is illegal to have two types with the same relative name within the context of the same web service, or within the context of any published web service.

Workaround: Use unique names.

Limitation in reporting when a typecode name contains a comma (CC-45305)

Issue: Due to the limitations of the InetSoft reporting software, issues arise when a passing a typelist as a parameter to a drill-down report if one of its typecode names contains a comma. (For example, typelist Loss Cause contains a typecode with the name *Fall, slip, or trip injury*.) If a typecode name does contain a comma, InetSoft thinks it is an array and sends multiple values. Guidewire has implemented a solution for the Loss Cause typelist in ClaimCenter. However, if you are passing any other typelist as a parameter to a drill-down report and it contains a typecode with a comma, you will experience this issue.

Workaround: Guidewire is aware of this issue and will address it in a future release. If you encounter this issue with any other report parameters, contact Guidewire Support for assistance.

Cannot upgrade to ClaimCenter 5.0.3 if you implemented archiving in an earlier release (CC-45993)

Issue: If you are using the archiving feature in ClaimCenter 5.0.0 through 5.0.2, you cannot upgrade to ClaimCenter 5.0.3.

Workaround: Contact Guidewire Support for assistance. Guidewire recommends using archiving from CC 5.0.5 onwards.

Typelist error message: "contains a typecode with an empty code/name" (CC-46085)

Issue: When starting the server, you may see an error that says a typelist contains a typecode with an empty code or name. This error is mistakenly generated and does not indicate any problem.

Workaround: Ignore this error.

Upgrade fails when upgrading CoverageType typelist that already has a code of: BLDG (CC-46510)

Issue: During upgrade, you receive an exception for the following upgrade trigger: `RecodeCoverageTypecodesTriggers$RecodeCoverageTypecode` and your Coverage Type typelist already has a typekey with code =BLDG

Workaround: Contact Guidewire Support for assistance.

Deny Check associated with a bulk invoice item is not supported (CC-46653)

Issue: Denying a check associated with a Bulk Invoice item (`check.Bulked == true`) results in an error. The Deny Check feature does not support such checks.

Workaround: Such checks share the same characteristics as the Bulk Invoice or Bulk Invoice item, so in the Bulk Invoice Validation plugin, check for the conditions that could lead to Denial downstream.

Unable to subclass `gw.api.quickjump.SpecificClaimCommand` (CC-49433)

Issue: If you extend the `gw.api.quickjump.SpecificClaimCommand` class to override the `isPermitted()` method, then the Specific Claim QuickJump command is unavailable. It also can cause an error during server startup.

Workaround: You can create a new QuickJump command that subclasses `com.guidewire.pl.web.navigator.commands.DefaultQuickjumpCommand`, and which delegates its operations to an instance of the `gw.api.quickjump.SpecificClaimCommand`. . If you need to create a `SpecificClaimCommand`, contact Guidewire Support for sample code attached to issue PL-4061.

Iterator buttons not working correctly in `MatterDetailsDV.pcf` (CC-48879)

Issue: On the `MatterDetailsDV` file, the **Add** button does not hide when not in *Edit* mode. Also, the **Add** and **Remove** buttons do not show when in *Edit* mode, on the `StatusLinesLV` file.

Workaround: Set `lockWhileEditing="false"`. Note it is not locked.

`ShowNewExposureChooseByCoverageMenuForLossTypes` parameter does not allow a blank value (CC-49426)

Issue: If the `ShowNewExposureChooseByCoverageMenuForLossTypes` configuration parameter is not defined in the `config.xml` file, then the server throws an exception trying to look for at least one loss type in this parameter. This is problematic if you do not want the menu to be displayed.

Workaround: Remove the menu from the PCF files.

FNOL wizard not saving selected vehicle (CC-47439)

Issue: The second step of the FNOL wizard does not save the selected vehicle when viewing the **Policy Details** screen and 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)." .`

Integrity checks allow offsetting payments to be associated to a different check than the payment it offsets (CC-49866)

Issue: Integrity checks for staging tables do not ensure that the two payments pointed to by a `TransactionOffsetOnset` row belong to the same check.

Workaround: You must correct the errors in data mapping, and ensure that each `Voided` or `Stopped` payment on a `Voided` or `Stopped` check also has an offset payment (with negative amount, in `Submitted` status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

Exception stack trace in user interface when `DisplayableException` thrown from Policy plugin (CC-47987)

Issue: A `GScript` util class throws a `DisplayableException` to the policy plugin `GScript` class which is again throwing another `DisplayableException` with the message from the first exception. The exception message is not user friendly.

Workaround: Add the following parameter to your `config.xml` file:

```
<param name="IntegratedStackTraces" value="true"/>
```

`ArithmeticException` when exchange rate is set to 0 (CC-48438)

Issue: When you enter an exchange rate = 0 in the New Check wizard, you see an `ArithmeticException` error in the user interface.

Workaround: Add a validation expression (attribute `validationExpression` or `requestValidationExpression`) to the `Transaction_ExchangeRate` element in `ExchangeRateInputSet.default.pcf` and `ExchangeRateInputSet.Check.pcf` files.

Sessions which have idled and been reaped still appear as active (PL-1699)

Issue: When a session times out, it still appears as active. This can be seen in the Management Bean for viewing active user sessions in the system.

Workaround: None. Guidewire is aware of the issue and will address it in a future release.

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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 info, 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.

Error when removing claim flags through the Team tab (CC-50159)

Issue: The code to remove the flagged status from a claim in the Team view only works with one claim at a time. The following PCFs allow you to select multiple claims in the LV and then click **Remove Flag**, which results in an error. This can happen with these PCFs: `TeamGroupOtherClaims.pcf`, `TeamGroupPendingClaims.pcf`, `TeamUserClaims.pcf`

Workaround: Modify the `flags` attribute of the Remove Flag button in each PCF to read:

```
flags="one CanRemoveFlag"
```

Cannot archive claims that are referenced by BulkInvoiceItem (CC-51808)

Issue: You cannot archive a claim that is referenced by `BulkInvoiceItem`.

Workaround: Do not attempt to archive claims that are referenced by a `BulkInvoice` or `BulkInvoiceItem`, until this issue is fixed. An archive rule has been added to the ClaimCenter 5.0.5 configuration which prevents those claims from being archived. Guidewire strongly recommends leaving this rule in place and, if required, merge it into your existing configuration. Guidewire intends to fix this issue in a future maintenance release.

Group cache needs correct sizing (CC-51545)

Issue: Guidewire discovered assignment performance issues during the ClaimCenter 5.0.5 performance testing on a 2.5M claim database due to the fact that the default configuration of the `GroupCache` was not large enough to hold all the groups. This resulted in frequent premature evictions from the cache during assignment and subsequent thrashing of the cache.

Workaround: Increasing the cache size to make it large enough to hold all the groups (2200 in this case) significantly helps assignment performance. If possible, Guidewire recommends that you configure the value for the `GroupCache` size (located in the `config.xml` file) to make it large enough to hold all groups.

When editing a check, check portions should not exceed total payments (CC-45520)

Issue: In the **Edit Check** wizard, the fixed-amount check portions are not prevented from exceeding the total of the payments on the multi-payee check.

Workaround: You must make the following configuration change in the `EditCheckWizard.pcf` file: add the method `Wizard.validatePayments` to the `onExit` expression for the *NewCheckPayments* wizard step.

Metropolitan Police reports are receiving duplicate requests (CC-51147)

Issue: ClaimCenter is sending duplicate requests to Metropolitan Police reports which results in the customer getting billed for each request. This happens because the order request templates has the `ForceDup` set to `Yes`.

Workaround: The workaround is to create a new field on the `MetroReport` entity to specify whether a report should be requested from Metropolitan regardless if a duplicate request has been made.

1. Using Studio, in each report template, change:

```
<mrp:ForceDup>Yes</mrp:ForceDup>
```

to

```
<mrp:ForceDup><%=metroReport.ForceDuplicate?"Yes":"No"%></mrp:ForceDup>
```

2. Add an extension field `ForceDuplicate` to the `Metro` entity by inserting the following into the `extensions.xml` file, by default we set this to `true`:

```
<extension entityName="MetroReport">
  <column name="ForceDuplicate" type="bit" default="true" desc="Whether a metro report is requested
  regardless of a duplicate request."/>
</extension>
```

3. Modify the Metropolitan Report user interface to take `ForceDuplicate` by inserting the following into the `MetroReportDetailsScreen.pcf` file:

```
<Input label="Force Duplicate" id="ForceDuplicate" value="MetroReport.ForceDuplicate"
editable="true" />
```

Problematic calls to SOAP API methods (CC-45378)

Issue: Calling ClaimCenter SOAP API methods from within a messaging plugin can be problematic because the root entity is locked during message processing.

Workaround: If you are upgrading to 5.0.x and your 4.0.x or earlier configuration is calling SOAP API functions from within messaging plugins (or message sinks), you need another approach. Contact Guidewire Support for information on alternatives to calling SOAP APIs from within your messaging integration code.

Error when removing claim flags through the Team tab (CC-50160)

Issue: The code to remove the flagged status from a claim in the *Team* view only works with one claim at a time. The following `.pcf` files allow you to select multiple claims in the list view and then click **Remove Flag**, which results in an error. This can happen with these `.pcf` files: `TeamGroupOtherClaims.pcf`, `TeamGroupPendingClaims.pcf`, `TeamUserClaims.pcf`.

Workaround: Modify the `flags` attribute of the **Remove Flag** button in each `.pcf` file to read:

```
flags="one CanRemoveFlag"
```

Invalid activity pattern referenced in a workflow prevents ClaimCenter from starting (PL-5987)

Issue: If a workflow configuration file has an `ActivityStep` that references an activity pattern that does not yet exist in the system or database, (the activity pattern could be created after deployment in a real production release), then the application fails to start.

Workaround: Do the following:

- Remove all the references to the workflow activity pattern
- Deploy
- Create the activity pattern
- Re-deploy the application with the references to the workflow activity pattern in place.

Duplicate descriptions for image/jpg mime types in config.xml (CC-50371)

Issue: In the config.xml file, there are two entries under MIME types for jpeg: one for pjpeg and one for normal jpeg. This causes a problem as the descriptions of both are identical. If you upload a normal 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 Studio.

License state drop down menu in the user interface produces unexpected results (CC-50529)

Issue: If you use the AddressAutocompleteUtil.getStates() method to populate license states in the FNOLVehicleIncidentPopup.pcf file, then you might see results that are not expected, such as countries or retired typecodes. This occurs when you select the license state drop down menu in the user interface. What is happening is that AddressAutocompleteUtil.getStates is passing true to the getTypeKeys method when it should be false.

Workaround: Use the following:

```
function getStates_Ext(country : typekey.Country) : typekey.State[] {
    var stateList = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = "US"; //default to US
    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
            stateList.add(eachState)
        }
    }
    return stateList.toArray() as typekey.State[]
}
```

Include this function in your .pcf file. Instead of calling the AddressAutocompleteUtil method in the valueRange, call this method instead:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

Incorrect additivity statements in the default logging.properties file (CC-50615)

Issue: Some additivity statements in the logging.properties file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the logging.properties file (located at modules/cc/config/logging), change:

```
log4j.additivity.PluginsLog=false
```

to

```
log4j.additivity.Plugin=false
```

The statements for API, Database, Messaging, Messaging.ISO, Plugin.ClaimNumGenLog, Plugin.IPolicySearchAdapter, and Plugin.IContactSearchAdapter are also incorrect and need to be changed.

Claim association can result in data base consistency check failure (CC-50724)

Issue: Making a claim association from the user interface can result in making two claims as primary. This results in data base consistency check failure.

Workaround: In the user interface, edit associations and make *only one* claim as primary.

Paging controls are disabled while trying to reassign an activity (CC-50899)

Issue: In the user interface, the paging controls become disabled when you try to reassign an activity by searching for a queue.

Workaround: In Studio, set `startInEditMode=true` in the `AssigneePickerPopup`. This is located in the **Advanced Properties** section.

Visible property value is missing in `MatterDetailPage.pcf` (CC-50673)

Issue: In the `ClaimMatters.pcf` file, the `Assign` button has the visible property set to `perm.Matter.genericassign` and it will not display this button if you do not have the correct permission. However, this is not set in the `MatterDetailPage.pcf` file and it should be set there as well.

Workaround: Add `(perm.Matter.genericassign)` to the assign button's *Visible* property in the `MatterDetailPage.pcf` file.

Exposure level security causes error on Planned Activities in Claim Summary screen (CC-50696)

Issue: When exposure level security is enabled, if you do not have access to an exposure, then you should also not have access to view the exposure's activity detail. Yet in the **Claim Summary** screen → **Planned Activities** you can still click the link of the activity to view the details, even though you see the message: **You do not have the permission required to perform this action: ActivityDetailForward.**

Workaround: Either the Activity should not show in the planned activities list or the link should be disabled so users that do not have the correct security permissions cannot access it.

Duplicate check search velocity template is incorrect (CC-50688)

Issue: There is a bug in the Velocity template used to construct the query for finding duplicate checks in the check wizard. It does not handle all the cases where one of the `ServicePdStart/ServicePdEnd` fields is null on one check but not on the other. Usually this does not matter because both these fields are set together so they are both either set or not set, in which case you can use the existing template.

Workaround: Find the following lines in the `check.vm` file in Studio under **configuration** → **Other Resources** → **duplicate-search**. The template can be changed to be more consistent. Instead of:

```
#if ($Check.servicePdStart)
AND cc_check_head.ServicePdEnd > $Util.sql.getSQLStringValue($Check.servicePdStart)
#end
#if ($Check.servicePdEnd)
AND cc_check_head.ServicePdStart < $Util.sql.getSQLStringValue($Check.servicePdEnd)
#end
```

The template can be changed to:

```
#if ($Check.servicePdStart && $Check.ServicePdEnd)
AND cc_check_head.ServicePdEnd >= $Util.sql.getSQLStringValue($Check.servicePdStart)
AND cc_check_head.ServicePdStart <= $Util.sql.getSQLStringValue($Check.servicePdEnd)
#end
#if (!$Check.servicePdStart || !$Check.ServicePdEnd)
AND (cc_check_head.ServicePdStart IS NULL OR cc_check_head.ServicePdEnd IS NULL)
#end
```

The adjusted version compares check service periods if both start and end values are set. Otherwise, it assumes all service periods with either only one bound or neither bounds set match. So checks with complete service periods (both lower and upper bound set) match if the service periods overlap. Checks with incomplete service periods all match (at least as far as their service periods are concerned).

Since this template is configurable, you should not take these changes if you have already customized this logic to your business requirements. Also, you do not need to make these changes if you are satisfied with the current matching behavior.

Duplicate claim search velocity template is incorrect (CC-50686)

Issue: There is a bug in the released version of the `claim.vm` file, the template used to construct the duplicate claim query used in the New Claim wizard. The query has two parts:

- The first looks for a claim with the same policy number and a loss date within three days of the current claim. This part of the query is correct.
- The second looks for a claim with the same insured and a loss date within three days of the current claim. This part of the query is *incorrect*.

Workaround: Find the following lines in the `claim.vm` file in Studio under **configuration** → **Other Resources** → **duplicate-search**:

```
#if ($Util.getEntityIntrinsicType("Person").isAssignableFrom($Claim.insured.entityIntrinsicType))
    AND cc_contact_2.FirstNameDenorm = $Util.makeParam("Claim.Insured.Name",
    $Claim.insured.firstName)
    AND cc_contact_2.LastNameDenorm = $Util.makeParam("Claim.Insured.Name", $Claim.insured.lastName)
#else
    AND cc_contact_2.NameDenorm = $Util.makeParam("Claim.Insured.Name", $Claim.insured.name)
#end
```

Change them to:

```
#if ($Util.getEntityIntrinsicType("Person").isAssignableFrom($Claim.insured.entityIntrinsicType))
    AND cc_contact_2.FirstNameDenorm = $Util.makeParam("Claim.Insured.Name",
    $Claim.insured.firstNameDenorm)
    AND cc_contact_2.LastNameDenorm = $Util.makeParam("Claim.Insured.Name",
    $Claim.insured.lastNameDenorm)
#else
    AND cc_contact_2.NameDenorm = $Util.makeParam("Claim.Insured.Name", $Claim.insured.nameDenorm)
#end
```

Without these changes, the second part of the query will rarely match anything on a case sensitive database, because they are comparing a denormalized, lower case, value with a non normalized mixed case value.

Checks Search and Recoveries Search screens execute a query to populate the footer even when no selection is made (CC-50940)

Issue: On single-currency configurations, this change makes a workaround possible for the problem of the sum query being unnecessarily evaluated on the search screens.

Note: This workaround will not produce correct results for servers running in multi-currency mode.

Workaround: To implement this workaround, you must make the following .pcf file changes in Studio:

In the `BulkInvoiceSearchResultsLV.pcf` file:

```
On the "ApprovedAmount" Cell, replace this:
    footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.ApprovedSum) : null"
with this:
    footerSumValue="BulkInvoiceSearchView.ApprovedTransactionAmount"
On the "TotalAmount" Cell, replace this:
    footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.TotalSum) : null"
with this:
    footerSumValue="BulkInvoiceSearchView.TotalTransactionAmount"
```

In the `CheckSearchResultsLV.pcf` file:

On the "Amount" Cell, replace this:

```
footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.Sum) : null"
```

with this:

```
footerSumValue="CheckSearchView.GrossAmount"
```

In the `RecoverySearchResultsLV.pcf` file:

On the *Amount* Cell, replace this:

```
footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.Sum) : null"
```

with this:

```
footerSumValue="RecoverySearchView.Amount"
```

Unable to access GScript from Java class (CC-52191)

Issue: The documentation is incorrect regarding generating Java types in the Java external entities (the Java Entity Libraries) for arbitrary GScript classes. The application *only* generates Java classes for types that are entities, typelists, plugin interfaces, or a type attached to any of the previously mentioned objects because of properties or methods. So, if your GScript class is not referenced by an entity, typelist, or plugin, then ClaimCenter does not generate an external Java class for it.

Workaround: To expose some arbitrary GScript class, create a property or method on an entity using a GScript enhancement, declared to return an instance of your class. If you only use static methods on the class, then your implementation can return `null` since the declaration alone is sufficient to include that class in the Java libraries. If no built-in entity seems appropriate for your GScript enhancement, then create a new non-persistent entity in the data model configuration files. Next, create GScript enhancements for methods or properties to return an instance of your GScript class. After you regenerate the Java Entity Libraries, your GScript class is available from Java.

chapter 30

Guidewire ClaimCenter 5.0.6 Release Notes

Release 5.0.6.15

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Insurance Suite, Guidewire ContactCenter, Guidewire Studio, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries. All other product names are trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

- This release of Guidewire ClaimCenter is 5.0.6.15
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 9.0, build 20090817.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter

The following issues pertain to installing this release.

Inetsoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 9.0 using the following license key:

L000-75F-ERX-00009700100001F-F2AFD44C26F6

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide*.

Upgrade Issues

CC-56307: An upgrade step was created in order to remove the category element from the `ContactRole.xml` typelist file and to delete the previous `RoleConstraint.xml` typelist file.

Software Updates

There are several software updates that were implemented since 5.0.4:

- The release supports Microsoft Internet Explorer 8
- If you use SQL Server 2005, you must install its Service Pack 3 (SP3)
- You must run 64-bit operating systems on production application servers

Archiving Updates

Archiving in ClaimCenter 5.0.6 has undergone major structure changes. ClaimCenter archives claims using a *Distributed Work Queue*, which archives one claim at a time, as opposed to archiving multiple claims at once. Guidewire made these changes because it found that customer configuration of extension entities and fields could very likely break the archiving process. The new methodology adds additional verification of the data model, and ensures any problems will cause archiving a claim to fail early, rather than after a claim has already been archived (and so cannot be restored).

Guidewire has also included additional checks to be run on the claim graph. You can view the results of these checks by accessing the **Internal Tools** screen. While you might not go live with archiving in this release, Guidewire recommends that you resolve these warnings before going live on ClaimCenter 5.0.6.

See the following issues for additional details:

- PL-6082
- PL-6311
- CC-52441
- CC-52446
- CC-55010
- CC-55551
- CC-55753
- CC-56373
- CC-56661
- CC-56736
- CC-57115
- CC-57542
- CC-52744

Changes to HTML Tables

Internet Explorer's rendering of HTML tables may cause labels to wrap more aggressively than in previous ClaimCenter versions. The wrapping occurs most commonly in the left-most columns of detail views, but can appear in other places throughout the product interface. The issue was explained in the previous (ClaimCenter 5.0.4) release notes under the section *Wrapping of Labels*.

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes
- Base Resource Changes
- Improvements and General Issues

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 5.0.5 and ClaimCenter 5.0.6, *click here* (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- GScript API
- typelists
- display keys

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.5 to 5.0.6

- 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/platform` directory, [click here](#).

Base Resource Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.5 to 5.0.6

- To view a report of the changes in the base resources in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base resources in the `modules/platform` directory, [click here](#).

Improvements and General Issues

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

ID	Description
Associated Claims	
CC-53979	Fixed an issue where it was possible to create two claim associations with the same title if you committed them as part of the same transaction. This is an unlikely case and could never happen through the normal ClaimCenter user interface. However, it is possible you might have encountered it when writing custom user interface or integration code.
Administration	
CC-52744	Fixed an issue where the archive batch process would not start. Now that archiving feature has changed, the archive process is no longer a batch process, but a work queue. It does not have a Next Schedule Run column.
CC-55741	Fixed an issue where if you have a large admin tree, particularly if several nodes are open, the refresh time on the tree took a long time.
Aggregate Limits	
CC-52239	Fixed a potential performance issue that could occur when adding policy periods to a policy that was linked to a large number of claims.
Claim Archiving	
CC-52441	Corrected an issue where you were unable to delete a user who owned an archived claim.
CC-52446	Fixed an issue where a warning in the ClaimCenter user interface prevented the reopening of a claim that was retrieved from an archived database.
CC-55010	Since the process of archiving has changed, Guidewire has deprecated the <code>markForArchive</code> method since it is no longer necessary. The <code>scheduleForArchive</code> method is the closest thing to a replacement for the <code>markForArchive</code> method, and it takes either the numbers or public IDs of the claims to be archived. The JavaDoc has been expanded so that it explains the behavior of the calls and the possibility of a race condition with the admin data.
CC-55551	As part of the changes made to the archiving feature, specifically the Archive Work Queue, the <code>MarkedForArchive</code> typecode on <code>History</code> was removed.
CC-55753	Fixed an issue that involved archiving a claim against an unverified policy. The archive process failed with a null constraint violation. The violation occurred when the contact had a related contact that was not associated with the policy or claim directly.
CC-56373	Fixed an issue where restored claims did not get picked up for archive.

ID	Description
CC-56661	<p>Corrected an issue so that claims that can be purged can now be identified using a purge date. The fix included:</p> <ul style="list-style-type: none"> • the creation of a new nullable property called <code>PurgeDate</code> on the <code>Claim</code> and <code>ClaimInfo</code> entities • creation of a <code>Claim Closed</code> rule that sets the purge date to seven years after the claim close date • creation of a <code>Claim Reopen</code> rule that clears the purge date <p>Both rules are disabled in the base configuration. However, if you do decide to enable them, you must enable <i>both</i> of them.</p>
CC-56736	<p>Corrected an issue where ClaimCenter generated an exception in the user interface if you tried to transfer a check or recovery to an archived claim. The fix included removing the option in the user interface so that you cannot search for an archived claim.</p> <p>The affected files are:</p> <pre>/config/web/pcf/search/claims/ClaimSearch.pcf /config/web/pcf/search/claims/ClaimSearchPopup.pcf /config/web/pcf/search/claims/ClaimSearchScreen.pcf</pre>
CC-57115	<p>The update Archive batch process name and description have been changed to reflect the new archiving framework.</p>
Assignment / Segmentation / Strategy	
CC-55102	<p>Added the new method <code>getAllRoleAssignments()</code> in <code>Claim</code>. This method augments the method <code>getRoleAssignments()</code> which filters out any <code>UserRoleAssignment</code> that has an <code>ExposerID</code>. The new method returns all role assignments.</p>
New Claim Wizard	
CC-50686	<p>The Duplicate Claim Search Velocity template contained an incorrect query which was fixed. The fix now correctly searches for a claim with the same insured and a loss date within three days of the current claim.</p>
CC-56892	<p>Clarified the documentation for the <code>FirstAndFinal</code> property on the <code>Claim</code> entity to indicate that it is not saved to the database, and its value is only valid during the FNOL wizard.</p>
Core	
PL-3488	<p>Guidewire has modified <code>build.xml</code> to support user-specific versions of the <code>build.properties</code> file. Within a multi-user development environment, you can now create your own <code><user.name>.build.properties</code> file alongside the shipped <code>build.properties</code> file. As long as <code><user.name></code> matches your system user ID, it will be processed in a way that it overrides base Ant properties.</p>
PL-5603	<p>Corrected an issue that caused an exception on the (System Tools) Archive Info page that occurred if archiving was not enabled.</p>
PL-6082	<p>Guidewire has added an Archive Graph Info page to the Archive Info tool. The new page provides access to DOT format representations of the domain and admin graphs and any associated violations of warning-level checks.</p>
PL-6124	<p>Corrected an error that caused a Null Pointer Exception during a database rollback of a new array member that had been rekeyed. This only occurs under very rare circumstances in which the validation rules throw exceptions.</p>
PL-6245	<p>With this release, adding <code>?stateful</code> to the end of a Guidewire web service call URL will cause an application server session to be created, along with the usual cookies. The session cookie can be used by load balancers for session stickiness.</p>
PL-6311	<p>Corrected an issue where entities with references to <code>Contacts</code> generated foreign key constraint violations during archiving.</p>
PL-7333	<p>Corrected an issue with the <code>MessageTransport</code> plugin that caused it to not shut down if you shut down the application server.</p>
PL-7927	<p>Corrected an issue where the IDs of retired entities were being set to zero as they were archived.</p>
PL-7937	<p>Guidewire has made improvements to the look of the domain graph to make it more readable and consistent.</p>
PL-8205	<p>Commons logging was updated from version 1.0.4 to version 1.1.1</p>

ID	Description
PL-8636	<p>Guidewire has added several new scheduled batch processes and configuration parameters that work in conjunction to delete table entries and logs for completed workflow that are older than N days.</p> <p>The scheduled batch processes are:</p> <ul style="list-style-type: none"> • <code>PurgeWorkflowLogs</code> • <code>PurgeWorkflows</code> <p>The new configuration parameters are:</p> <ul style="list-style-type: none"> • <code>WorkflowLogPurgeDays01d</code> (If running the purge workflow process, the maximum age of the kept non-failed workflows) • <code>WorkflowPurgeDays01d</code> (If running the purge workflow logs process, the maximum age of the kept log entries for the non-failed workflows)
PL-8638	Corrected a messaging issue that reported and calculated the first time interval from 0 instead of the start time. This occurred if MESSAGING DEBUG was enabled.
PL-8759	<p>Guidewire has changed how it parses the digits entered by the user into a field. Previously, the algorithm (1) ignored any trailing zero(s) on entered numbers even if the number of entered digits exceeded the number set for that field, and (2) rejected decimal fields with no digits on the left side of the decimal point even though the scale setting was not violated (for example, ".00001" with scale = 5).</p> <p>Guidewire now takes trailing zeros into account. For example, US dollar currency amounts are allowed two digits after the decimal point. Previously, the application would accept \$1.230. With the change, this is no longer acceptable and the application flags that entry as an error.</p>
PL-8760	Corrected an issue with query generation that caused Oracle <code>SQLException: identifier is too long</code> . The exception occurred with queries in nested as a subselect of another query on extension tables that approached the 30 character table name limit. This error only occurs if you are using Oracle as the database.
PL-8946	Fixed an issue with imported webservises that caused an exception to be thrown when a single-entry array was returned from a web service call.
PL-9043	Corrected an issue that caused an error condition if two different users tried to log into the same application entry point concurrently.
PL-9215	Guidewire has added configuration parameter <code>ArchiveEnabled</code> that controls whether the application creates several indexes on certain archive-related tables.
PL-9292	Updated the joda-time library from version 1.4 to 1.6 to correct a problem with the IBM JDK with WebSphere that returned incorrect results for class <code>DateMidnight</code> .
PL-9688	Corrected an issue that occurred when running <code>maintenance_tools -markclaimsforpurge</code> against a claim with a large number of checks (4000). This caused a <code>SQLServerException</code> .

Database Upgrader

CC-53378	Corrected an issue where the <code>BodyPartDetailsExtensionVersionCheck</code> was failing in error.
----------	--

Documentation

CC-53300	The previous release note in version 5.0.5 incorrectly stated that the templates contain a <code><ForceDup></code> XML element, which is calculated by looking at the <code>MetroReport.ForceDuplicate</code> field. By default <code>ForceDuplicate</code> is false which results in <code><ForceDup>No</ForceDup></code> . The default value of the <code>ForceDuplicate</code> flag should be set to false so that you are not billed for duplicate orders.
CC-55369	The default currency symbol for InetSoft is now set as part of the configuration process through the <code>sree.properties</code> file. This applies to ClaimCenter 5.0.5 and future releases. See the Jira comments section for a description of the property settings.
CC-55730	Guidewire does not support the ability to make extension entities assignable. The documentation stated otherwise. This documentation has been corrected.

Financials

CC-46653	Corrected an issue where denying a bulk invoice check threw an exception in the user interface. While it is disallowed, now there is an appropriate message in the user interface. Only requested or requesting single-payee, non-bulk checks can be denied.
CC-47223	The description of properties <code>RateScale</code> and <code>NormalizedRate</code> on the <code>ExchangeRate</code> entity was updated to consider them deprecated. Use the <code>Rate</code> property instead to set exchange rates.
CC-49648	Modified the data model to fix an issue with the entity <code>BulkInvoiceItemInfo</code> in the archiving claim graph.
CC-50428	Clarified the Javadoc for the <code>TransactionSet.testAuthorityLimits()</code> method so that it specifies which user it tests.

ID	Description
CC-51264	Fixed an issue where the Payment Exceed Reserves authority limit did not correctly work.
CC-51902	Fixed an issue where the IClaimFinancialsAPI was allowing payments to be created with a NULL value for PaymentType, but that is now prevented. It should be partial, supplemental, or final.
CC-52573	Corrected an issue where a claim-level check that was part of a bulk invoice was being escalated when its claim was closed, even though the check had PendEscalationForBulk = true. Now closing the claim will not escalate the check.
CC-53234	Fixed an issue where the method setPaymentType() on the Payment entity now calls the method setAsNonEroding(), and does not set the DoesNotErodeReserves flag directly. This fix applies only for supplemental payments (payment type: supplement).
CC-53369	Fixed an issue where the system gave a NullPointerException and the bulk invoice item processing would stop if a bulk invoice was re-submitted. This is during the second time, after an initial problem, and so existing placeholder checks had to be updated if the bulk invoice payee was not linked to the Address Book (and did not have an AddressBookUID). Guidewire recommends that a bulk invoice's payee contact to be linked to the Address Book, although Guidewire is not enforcing this in 5.0.x. However, it will be enforced in versions 6.0.x.
CC-53522	Added the configuration parameter BulkInvoiceItemProcessorBatchSize, so you can see exactly which bulk invoice item and claim fails validation during the bulk invoice processing.
CC-53817	Corrected an issue where the search results used in a recovery search were inaccurate because the number of results came from the line item level and not the transaction level.
CC-53937	Fixed an issue where the total approved amount in the bulk invoice screen displayed incorrect amounts after editing the bulk invoice item's payment type.
CC-53943	Fixed an issue where the Checks list view displayed incorrect checkRpt values for an updated multiple line item payment. This issue existed only with multi line item payment update.
CC-54101	Fixed an issue where transferring a check from one claim to another would make it impossible for either claim to be archived. This is because there is a link (foreign key) from the original payment to the transferred payment on the new claim. ClaimCenter now manages this foreign key during archive and restore, so it does not interfere with archiving.
CC-55939	Fixed an issue where the system threw a NullPointerException in the user interface if the deduction's transactionAmount and ClaimAmount are both null.
CC-56183	Two new consistency checks were added to verify that: <ul style="list-style-type: none"> • Every bulk invoice's payee is linked to ContactCenter • No bulk invoice's payee is also a ClaimContact or otherwise linked to an extractable entity
CC-56345	Changed the PaymentType field on the Payment entity to be non-nullable. In the base configuration, a required value of either partial, supplement, or final must be selected in the user interface. The fix included having the version check verify that the PaymentType is not null.
CC-56418	Fixed an issue where the PaymentsExceedReservesSum class returned an incorrect result if it found a reserve line where the payments exceeded reserves but by less than the authority limit.
CC-56452	The sample rule for Manually Checking Authority Limits during the Transaction Approval process was changed to update the TransactionSet.RequestingUser before testing authority limits. This makes the rule behave as expected (just like automatic authority limit checking) when multiple approvals are required. This rule is disabled in the base configuration and only applicable if the CheckAuthorityLimits configuration parameter is set to false. If this applies to your configuration and you have enabled this rule, you must examine the changes made and perform testing after merging in the changes, since the change may modify approval behavior.
CC-56562	The links Open Claim Financials and Period Financials on the Dashboard tab in the user interface contain calculations of payments and recoveries, such as Total Recovered and Net Total Incurred. Those calculations should not include any offset payment or recovery transactions. The fix removed the offset transactions from the calculations.
CC-56674	Corrected an issue where the AbilityToPayResult class was returning messages as pure text strings. The fix now returns messages that can have title with links pointing to the places in the user interface that failed to validate.
CC-56698	Fixed an issue where you could not add and then immediately remove a document attachment to a check set before clicking Finish in the New Check wizard. This issue also occurred while creating an activity on a claim when you linked a document and then removed it from the list view and then saved the activity.

ID	Description
CC-57168	Fixed issues where various operations on a bulk invoice (such as submit, request, void/stop, and delete) or its items (edit, process, and delete) could throw an exception if the claim associated with a bulk invoice item was archived. Most of these operations are not possible until the archived claim is restored, so if this occurs, the system now shows a meaningful error message. You cannot edit a bulk invoice item in the user interface if its claim is archived.
CC-57220	Corrected an issue in InetSoft financial reports where non-eroding payments affected the remaining reserves on a claim. The fix included changing the calculation. The reports that were affected were: <ul style="list-style-type: none"> • Claim Injury Detail • Reopened Claims List • Large Loss Financials
Integration	
PL-2567	Corrected a problem with the display of Japanese characters. In some cases, the browser showed the field label in Japanese characters vertically instead of horizontally. This occurred if the user minimized the browser window beyond a certain amount.
PL-5611	Guidewire has improved the time required to render the Group/User tree on the ClaimCenter Team page.
PL-6097	Guidewire mistakenly removed button Check for Duplicate Contacts in ClaimCenter 5.0.5. It has been reintroduced in ClaimCenter 5.0.6.
PL-8089	Corrected issues with secondary assignment so that it works correctly outside of the Rules engine.
PL-9678	Corrected an issue that caused the Guidewire application to not be able to open an attached document if the document name contained a long sequence of Japanese characters.
Internationalization	
CC-55546	Fixed an issue in which a CSV export to Microsoft Excel using an English (UK) locale created an invalid character.
ISO Integration	
CC-53556	Corrected an issue so that Medicare information could be successfully sent to ISO. The fix included correcting the code that sends the payload to ISO. This means that elements with "." and "_" can now be sent successfully.
Manageability	
PL-7596	Guidewire has modified the behavior of the Sequence Generator to improve its performance, especially with work queues with multiple threads. The signature of <code>SequenceUtil.next()</code> method that has a <code>maxRetries</code> parameter has been deprecated in this release. Guidewire intends to remove this method signature in subsequent releases as that parameter is no longer necessary.
PL-7895	Guidewire has added the .html extension to HTML pages data distribution download .zip files. You can now click the file to open the file in a browser.
PL-8419	Guidewire has made improvements to the summary download file accessible from the (System Tools) Work Queue Info page. The file now provides details on the individual worker summaries, including actual execution time and throughput by total uptime and during actual execution.
PL-9177	Added ability to generate and download information on data distribution comparisons from the (Server Tools) Info Page → Data Distribution page.
PL-9197	Guidewire has added a new Info page to the System Tools available to administrators. Named Runtime Environment Info , it contains information on the Application Server versions, the Java Runtime version, and whenever possible, the hardware configuration.
Metropolitan Police Reports	
CC-54269	Corrected an issue where the Metropolitan police report workflow was caught in a loop if it did not get a reply from Metropolitan.
Reporting	

ID	Description
CC-56243	<p>Corrected an issue where post processing was affecting the performance of certain reports (using the integrated InetSoft Reporting software). The fix included rebuilding the tables on a single logical model in order to eliminate the log message of <i>Condition is not mergeable, run as post-processing</i>.</p> <p>Those reports included:</p> <ul style="list-style-type: none"> • Monthly Recoveries • Monthly Claim Costs Paid • Monthly Expenses and Claim Costs • Monthly Expenses Paid • Monthly Financial Transactions
CC-56244	<p>Corrected an issue where post processing was affecting the performance of certain reports (using the integrated InetSoft Reporting software). The fix included rebuilding a table on a single logical model in order to eliminate the log message of 'Condition is not mergeable, run as post-processing'.</p> <p>The report affected was Past 30 Day Claim Summary Report.</p>
CC-56246	<p>Corrected an issue where the InetSoft reporting software financial reports for Loss Run contained incorrect calculation which resulted in returning the wrong value for total incurred. The affected views were:</p> <ul style="list-style-type: none"> • ccrv_transaction • ccrv_transaction_metrics • ccrv_loss_run • ccrv_claim_financials • ccrv_claim_metrics <p>Reports with calculations (incurred and reserves) modified:</p> <ul style="list-style-type: none"> • Catastrophe Financials by Claim Owner • Claim Detail Coverage • Claim Injury Detail Number • Claims/Claim Injury Detail • Claims/Reopened Claims List • Dashboards/Open Claim Counts • Dashboards/Open Claim Financials Dashboard • Financials/Catastrophe Financials • Financials/Current Catastrophe Financials • Financials/Open Claim Financials
CC-57328	<p>Corrected an issue where the LOB, Loss Type, and Loss Cause parameters were not set in several reports in the InetSoft Report Portal and therefore the applicable claims did not display. The fix included using the en_US for the default locale if none was specified.</p>
Searching	
CC-51804	<p>Fixed an issue that occurred if you searched for an activity (through the Activity Search screen in the user interface) that contained a contact that was an external owner. ClaimCenter would display an error in the user interface if you navigated away from that screen and returned.</p>
Studio	
PL-6760	<p>Corrected an issue in which Studio corrupted rules if you attempted multiple actions to move rules around without waiting for the previous rule movement to complete. This occurred if you issued a second rule move command before the rule completely moved to its first destination.</p>
PL-7359	<p>Guidewire has corrected a problem with the PCF CurrencyValueWidget editor involving <i>Reflection</i>.</p>
Tool bug	
CC-52063	<p>Corrected an issue where the version control system would replace the tags \$Author, \$Revision, and \$DateTime with the version control system user name and date stamp. This broke the checksum.</p> <p>You can:</p> <ul style="list-style-type: none"> • Have the version control system ignore the keyword substitution, which can be configured at the repository level, or • Use the -ko option as you check in files which tells the version control system to not use keyword substitution on the file.
CC-57079	<p>The archive database configuration element was removed from the config.xml file since in the base configuration, archiving is turned off.</p>
Upgrade, Configuration	

ID	Description
CC-56307	Created an upgrade step to remove the category element from the ContactRole.xml typelist file and to delete the previous RoleConstraint.xml typelist file.
Web	
PL-8911	<p>In some cases, field labels in the application interface wrap aggressively creating a less-than-optimal look. To correct this problem, Guidewire recommends that you uncomment the following style in the application CSS file and set the width parameter appropriately.</p> <pre> /* default width for input label: */ .dv .inputLabel *, .dv .inputLabel_bold * { /*width: 180px;*/ } </pre>
PL-9249	<p>Guidewire discovered a security vulnerability in a previous ClaimCenter release in which a user who carefully manipulated a server request was able to view more data than that to which the user had legitimate access. This vulnerability was viewed on the Common Vulnerability Scoring System (CVSS) as Medium, with a low likelihood of discovery.</p> <p>Guidewire has modified the application to detect this kind of malformed request and to return permitted data only. Guidewire has incorporated the fix into this release and into all future releases.</p>

Known Issues and Limitations

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

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 login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER/Software/Microsoft/Windows/CurrentVersion/Explorer/BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situa-

tion, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands in UNIX, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in `filename`.

If you are using Windows, you can use Cygwin.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` method in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a Pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaimAPI depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: `IClaimAPI` depends on `SynchStateData`, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

An exception is thrown when trying to link a new contact to the address book from the new check wizard (CC-32732)

Issue: If you create a new contact in the New Check Wizard, and later attempt to link contacts to ContactCenter, you get an error.

Workaround: None.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To retrieve the payment's send date, both the Transactions and the Checks table have to be searched; this hampers performance and is not done.

Workaround: None.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The Reset button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the Reset button an Initialize button.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you access the ClaimCenter Desktop Activities page and repeatedly click Activities, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to either Internet Explorer 7 or 8.

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the Plan of Action → New Evaluation page will display an incorrect title *New Evaluation* instead of *LOBName Evaluation*, and the page will not have a *LOBName Cost* field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use ClaimEvaluationDetail page instead of ClaimEvaluationDetails<LOB> page. Also, do not customize the ClaimEvaluationDetail page heavily.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in SampleAcrobat.pdf.descriptor and SampleAcrobat.pdf files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

After upgrading from ClaimCenter 3.1, the doctor is no longer shown in medical detail exposures (CC-40595)

Issue: ClaimCenter 3.1 displayed the value Exposure.Doctor, but later versions of ClaimCenter display the value Claim.FirstIntakeDoctor. During upgrade, the data is not migrated from the old field to the new field, resulting in the upgraded ClaimCenter version showing a blank instead of the real value.

Workaround: Add another PCF field in ClaimCenter 5.0.x on the Medical Details page below the current Doctor field for Exposure.Claim.FirstIntakeDoctor. This displays the upgraded data.

Potential to lose LocationXXX data during upgrade (CC-41555)

Issue: Guidewire changed the location of the LocationCity, LocationState, LocationStreet, and LocationZip fields between ClaimCenter 3.1.x, 4.0.x, 5.0.0, and 5.0.1:

- In ClaimCenter 3.1.x, these fields live on the exposure.
- In ClaimCenter 4.0.x, these fields live as extensions on MobilePropertyIncident.
- In ClaimCenter 5.0.0, Guidewire moved these fields into a separate LocationAddress object, in the core data model.
- In ClaimCenter 5.0.1, Guidewire changed LocationAddress into an extension on MobilePropertyIncident (in extensions.xml).

Upgrading from ClaimCenter 3.1.x or 4.0.x. Guidewire added a version check that should prevent data loss when upgrading to 5.0.2 from either 4.0.x or 3.1.x. It will look for the LocationXXX fields in incident or exposure (depending on whether you are upgrading from 4.0.x or 3.1.x). If any of those fields exist, then the version check will insist the following:

1. There is a LocationAddress field, of type Address, somewhere in the incident hierarchy
2. That any incidents/exposures with the LocationXXX fields set map to incident types that contain the LocationAddress field. That is, it checks that all LocationXXX fields that contain a value have a LocationAddress field to use in migrating the data.

If either of these checks fail, the version check will abort the upgrade and you must then follow the steps outlined in the workaround to solve the problem.

Upgrading from ClaimCenter 5.0.0. If you are upgrading from ClaimCenter 5.0.0, there is already a LocationAddress field. However, in 5.0.0, Guidewire placed the field in the core data model, while in 5.0.1 and later it is in extensions.xml. So, if your target version extensions.xml does not contain a LocationAddress field, then the LocationAddress field could get dropped entirely during the upgrade process, possibly losing data. This is the case that the upgrade version trigger does not address.

Upgrading from ClaimCenter 5.0.1 or later. This is not an issue.

Workaround: Depending on the release from which you are upgrading, you need to do the following:

- **ClaimCenter 3.1.x base release.** In 3.1.x, the LocationXXX fields were in the base data model, on exposure. As part of the upgrade, you will need to decide which exposure types map to which incident subtype. You must map any exposure types that have the LocationXXX fields set to an incident subtype that has the LocationAddress field. This ensures that the upgrade process has a place to move the data.
- **ClaimCenter 4.0.x base release.** In 4.0.x, the LocationXXX fields were in extensions.xml, in the incident hierarchy. You could choose where in the incident hierarchy to put these fields. In upgrading to any 5.0.x version, replace these fields in extensions.xml with LocationAddress, at the same level in the hierarchy.
- **ClaimCenter 5.0.0 base release.** In 5.0.0, the LocationAddress field was in the base data model, in MobilePropertyIncident. If you are upgrading to 5.0.x, you will need to make sure LocationAddress is in the extensions.xml file. Guidewire recommends that you put it in MobilePropertyIncident. However, if desired, it is possible—because it now lives in extensions.xml—to move it to a different place in the incident hierarchy.

Transaction runWithNewBundle Does Not Follow Links (PL-1716)

Issue: The GScript API that runs a block of code with new bundle (gw.transaction.Transaction.runWithNewBundle(...)) has some limitations. Even if you call newbundle.add(yourentity), the add method does not follow the graph links to other entities to which it links. You must manually add each sub-object to the new bundle, using the call bundle.add(linkedEntity) for each entity. This includes all foreign key references to other entities as defined in the built-in data model files, but also includes customer data model extensions that include foreign key references.

Workaround: You can work around this by committing all entities in the current bundle instead of using this API. This may be an issue when designing implementing a custom web service. Committing all entities in the current bundle includes all entities serialized into the bundle as part of the web service request because they were parameters to a web service method. There may be cases where entities that are web service API parameters you do **not** want commit to the database.

One workaround is to redesign the web service API so instead of taking entities (the entities you do not want to commit) it takes GScript class instances instead of Guidewire entities. Your custom GScript classes can provide only the fields that you want to pass in your integration code, which may be a much smaller subset of fields in some cases. Because they are GScript classes and not entities, committing the entire bundle with `gw.transaction.Transaction.getCurrent().commit()` will skip that data that you choose not to commit to the database.

Creating a new document from a template results in a `ClassNotFoundException` (CC-42867)

Issue: If you implement the `IDocumentProduction` plugin in Java, then when you create a new document from a template you may encounter a `ClassNotFoundException`.

Workaround: Copy the file `ClaimCenter/modules/cc/config/plugin/ExternalMappings.properties` to `ClaimCenter/modules/configuration/config/plugin`. Then add the following line to the copied file:

```
gw.api.document.DocumentCreationInfo = gw.api.document.DocumentCreationInfo
```

In addition, copy the file `ClaimCenter/modules/cc/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf` to `ClaimCenter/modules/configuration/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf`. In the copied file, replace the following line:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.DocumentTemplateDescriptor
      .TemplateId, gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

with the following:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    util.document.DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.
      DocumentTemplateDescriptor.TemplateId,
      gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

After upgrade, two duplicate incidents are created (CC-44646)

Issue: If you have two exposures with the same incident type details (for example, auto damage on the same car with the same driver), the database upgrader creates two identical incidents.

Workaround: Please contact Guidewire Support for assistance.

Type name conflict in web services (CC-45132)

Issue: It is illegal to have two types with the same relative name within the context of the same web service, or within the context of any published web service.

Workaround: Use unique names.

Limitation in reporting when a typecode name contains a comma (CC-45305)

Issue: Due to the limitations of the InetSoft reporting software, issues arise when a passing a typelist as a parameter to a drill-down report if one of its typecode names contains a comma. (For example, typelist Loss Cause contains a typecode with the name *Fall, slip, or trip injury*.) If a typecode name does contain a comma, InetSoft

thinks it is an array and sends multiple values. Guidewire has implemented a solution for the Loss Cause typelist in ClaimCenter. However, if you are passing any other typelist as a parameter to a drill-down report and it contains a typecode with a comma, you will experience this issue.

Workaround: Guidewire is aware of this issue and will address it in a future release. If you encounter this issue with any other report parameters, contact Guidewire Support for assistance.

Typelist error message: "contains a typecode with an empty code/name" (CC-46085)

Issue: When starting the server, you may see an error that says a typelist contains a typecode with an empty code or name. This error is mistakenly generated and does not indicate any problem.

Workaround: Ignore this error.

Upgrade fails when upgrading CoverageType typelist that already has a code of: BLDG (CC-46510)

Issue: During upgrade, you receive an exception for the following upgrade trigger: `RecodeCoverageTypecodesTriggers$RecodeCoverageTypecode` and your Coverage Type typelist already has a typekey with code =BLDG

Workaround: Contact Guidewire Support for assistance.

Unable to subclass `gw.api.quickjump.SpecificClaimCommand` (CC-49433)

Issue: If you extend the `gw.api.quickjump.SpecificClaimCommand` class to override the `isPermitted()` method, then the Specific Claim QuickJump command is unavailable. It also can cause an error during server startup.

Workaround: You can create a new QuickJump command that subclasses `com.guidewire.pl.web.navigator.commands.DefaultQuickjumpCommand`, and which delegates its operations to an instance of the `gw.api.quickjump.SpecificClaimCommand`. If you need to create a `SpecificClaimCommand`, contact Guidewire Support for sample code attached to issue PL-4061.

Iterator buttons not working correctly in `MatterDetailsDV.pcf` (CC-48879)

Issue: On the `MatterDetailsDV` file, the **Add** button does not hide when not in *Edit* mode. Also, the **Add** and **Remove** buttons do not show when in *Edit* mode, on the `StatusLinesLV` file.

Workaround: Set `lockWhileEditing="false"`. Note it is not locked.

`ShowNewExposureChooseByCoverageMenuForLossTypes` parameter does not allow a blank value (CC-49426)

Issue: If the `ShowNewExposureChooseByCoverageMenuForLossTypes` configuration parameter is not defined in the `config.xml` file, then the server throws an exception trying to look for at least one loss type in this parameter. This is problematic if you do not want the menu to be displayed.

Workaround: Remove the menu from the PCF files.

FNOL wizard not saving selected vehicle (CC-47439)

Issue: The second step of the FNOL wizard does not save the selected vehicle when viewing the **Policy Details** screen and 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)"`.

Integrity checks allow offsetting payments to be associated to a different check than the payment it offsets (CC-49866)

Issue: Integrity checks for staging tables do not ensure that the two payments pointed to by a `TransactionOffsetOnset` row belong to the same check.

Workaround: You must correct the errors in data mapping, and ensure that each Voided or Stopped payment on a Voided or Stopped check also has an offset payment (with negative amount, in Submitted status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

Exception stack trace in user interface when `DisplayableException` thrown from Policy plugin (CC-47987)

Issue: A `GScript` util class throws a `DisplayableException` to the policy plugin `GScript` class which is again throwing another `DisplayableException` with the message from the first exception. The exception message is not user friendly.

Workaround: Add the following parameter to your `config.xml` file:

```
<param name="IntegratedStackTraces" value="true"/>
```

ArithmeticException when exchange rate is set to 0 (CC-48438)

Issue: When you enter an exchange rate = 0 in the New Check wizard, you see an `ArithmeticException` error in the user interface.

Workaround: Add a validation expression (attribute `validationExpression` or `requestValidationExpression`) to the `Transaction_ExchangeRate` element in `ExchangeRateInputSet.default.pcf` and `ExchangeRateInputSet.Check.pcf` files.

Sessions which have idled and been reaped still appear as active (PL-1699)

Issue: When a session times out, it still appears as active. This can be seen in the Management Bean for viewing active user sessions in the system.

Workaround: None. Guidewire is aware of the issue and will address it in a future release.

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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 info, 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 `checkForDuplications` in the `NewClaimWizard.pcf` file as seen in the following example:

```
function checkForDuplications() {
    if (Wizard.checkForNewDuplicateClaims()) { NewClaimDuplicationsWorksheet.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="checkForDuplications()"
    ...
```


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.

Error when removing claim flags through the Team tab (CC-50159)

Issue: The code to remove the flagged status from a claim in the Team view only works with one claim at a time. The following PCFs allow you to select multiple claims in the LV and then click **Remove Flag**, which results in an error. This can happen with these PCFs: `TeamGroupOtherClaims.pcf`, `TeamGroupPendingClaims.pcf`, `TeamUserClaims.pcf`

Workaround: Modify the `flags` attribute of the Remove Flag button in each PCF to read:

```
flags="one CanRemoveFlag"
```

Group cache needs correct sizing (CC-51545)

Issue: Guidewire discovered assignment performance issues during the ClaimCenter 5.0.5 performance testing on a 2.5M claim database due to the fact that the default configuration of the `GroupCache` was not large enough to hold all the groups. This resulted in frequent premature evictions from the cache during assignment and subsequent thrashing of the cache.

Workaround: Increasing the cache size to make it large enough to hold all the groups (2200 in this case) significantly helps assignment performance. If possible, Guidewire recommends that you configure the value for the `GroupCache` size (located in the `config.xml` file) to make it large enough to hold all groups.

When editing a check, check portions should not exceed total payments (CC-45520)

Issue: In the **Edit Check** wizard, the fixed-amount check portions are not prevented from exceeding the total of the payments on the multi-payee check.

Workaround: You must make the following configuration change in the `EditCheckWizard.pcf` file: add the method `Wizard.validatePayments` to the `onExit` expression for the `NewCheckPayments` wizard step.

Problematic calls to SOAP API methods (CC-45378)

Issue: Calling ClaimCenter SOAP API methods from within a messaging plugin can be problematic because the root entity is locked during message processing.

Workaround: If you are upgrading to 5.0.x and your 4.0.x or earlier configuration is calling SOAP API functions from within messaging plugins (or message sinks), you need another approach. Contact Guidewire Support for information on alternatives to calling SOAP APIs from within your messaging integration code.

Invalid activity pattern referenced in a workflow prevents ClaimCenter from starting (PL-5987)

Issue: If a workflow configuration file has an `ActivityStep` that references an activity pattern that does not yet exist in the system or database, (the activity pattern could be created after deployment in a real production release), then the application fails to start.

Workaround: Do the following:

- Remove all the references to the workflow activity pattern

- Deploy
- Create the activity pattern
- Re-deploy the application with the references to the workflow activity pattern in place.

Duplicate descriptions for image/jpg mime types in config.xml (CC-50371)

Issue: In the `config.xml` file, there are two entries under MIME types for jpeg: one for pjpeg and one for normal jpeg. This causes a problem as the descriptions of both are identical. If you upload a normal 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 Studio.

License state drop down menu in the user interface produces unexpected results (CC-50529)

Issue: If you use the `AddressAutocompleteUtil.getStates()` method to populate license states in the `FNOLVehicleIncidentPopup.pcf` file, then you might see results that are not expected, such as countries or retired typecodes. This occurs when you select the license state drop down menu in the user interface. What is happening is that `AddressAutocompleteUtil.getStates` is passing `true` to the `getTypeKeys` method when it should be `false`.

Workaround: Use the following:

```
function getStates_Ext(country : typekey.Country) : typekey.State[] {
    var stateList = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = "US"; //default to US
    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
            stateList.add(eachState)
        }
    }
    return stateList.toArray() as typekey.State[]
}
```

Include this function in your `.pcf` file. Instead of calling the `AddressAutocompleteUtil` method in the `valueRange`, call this method instead:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

Incorrect additivity statements in the default logging.properties file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false
```

to

```
log4j.additivity.Plugin=false
```

The statements for `API`, `Database`, `Messaging`, `Messaging.ISO`, `Plugin.ClaimNumGenLog`, `Plugin.IPolicySearchAdapter`, and `Plugin.IContactSearchAdapter` are also incorrect and need to be changed.

Claim association can result in data base consistency check failure (CC-50724)

Issue: Making a claim association from the user interface can result in making two claims as primary. This results in data base consistency check failure.

Workaround: In the user interface, edit associations and make *only one* claim as primary.

Paging controls are disabled while trying to reassign an activity (CC-50899)

Issue: In the user interface, the paging controls become disabled when you try to reassign an activity by searching for a queue.

Workaround: In Studio, set `startInEditMode=true` in the `AssigneePickerPopup`. This is located in the **Advanced Properties** section.

Visible property value is missing in `MatterDetailPage.pcf` (CC-50673)

Issue: In the `ClaimMatters.pcf` file, the *Assign* button has the visible property set to `perm.Matter.genericassign` and it will not display this button if you do not have the correct permission. However, this is not set in the `MatterDetailPage.pcf` file and it should be set there as well.

Workaround: Add `(perm.Matter.genericassign)` to the assign button's *Visible* property in the `MatterDetailPage.pcf` file.

Exposure level security causes error on Planned Activities in Claim Summary screen (CC-50696)

Issue: When exposure level security is enabled, if you do not have access to an exposure, then you should also not have access to view the exposure's activity detail. Yet in the **Claim Summary** screen → **Planned Activities** you can still click the link of the activity to view the details, even though you see the message: **You do not have the permission required to perform this action: ActivityDetailForward.**

Workaround: Either the Activity should not show in the planned activities list or the link should be disabled so users that do not have the correct security permissions cannot access it.

Duplicate check search velocity template is incorrect (CC-50688)

Issue: There is a bug in the Velocity template used to construct the query for finding duplicate checks in the check wizard. It does not handle all the cases where one of the `ServicePdStart/ServicePdEnd` fields is null on one check but not on the other. Usually this does not matter because both these fields are set together so they are both either set or not set, in which case you can use the existing template.

Workaround: Find the following lines in the `check.vm` file in Studio under **configuration** → **Other Resources** → **duplicate-search**. The template can be changed to be more consistent. Instead of:

```
#if ($Check.servicePdStart)
AND cc_check_head.ServicePdEnd > $Util.sql.getSQLStringValue($Check.servicePdStart)
#end
#if ($Check.servicePdEnd)
AND cc_check_head.ServicePdStart < $Util.sql.getSQLStringValue($Check.servicePdEnd)
#end
```

The template can be changed to:

```
#if ($Check.servicePdStart && $Check.ServicePdEnd)
AND cc_check_head.ServicePdEnd >= $Util.sql.getSQLStringValue($Check.servicePdStart)
AND cc_check_head.ServicePdStart <= $Util.sql.getSQLStringValue($Check.servicePdEnd)
#end
#if (!$Check.servicePdStart || !$Check.ServicePdEnd)
AND (cc_check_head.ServicePdStart IS NULL OR cc_check_head.ServicePdEnd IS NULL)
#end
```

The adjusted version compares check service periods if both start and end values are set. Otherwise, it assumes all service periods with either only one bound or neither bounds set match. So checks with complete service periods (both lower and upper bound set) match if the service periods overlap. Checks with incomplete service periods all match (at least as far as their service periods are concerned).

Since this template is configurable, you should not take these changes if you have already customized this logic to your business requirements. Also, you do not need to make these changes if you are satisfied with the current matching behavior.

Checks Search and Recoveries Search screens execute a query to populate the footer even when no selection is made (CC-50940)

Issue: On single-currency configurations, this change makes a workaround possible for the problem of the sum query being unnecessarily evaluated on the search screens.

Note: This workaround will not produce correct results for servers running in multi-currency mode.

Workaround: To implement this workaround, you must make the following .pcf file changes in Studio:

In the BulkInvoiceSearchResultsLV.pcf file:

```
On the "ApprovedAmount" Cell, replace this:
    footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.ApprovedSum)
    : null"
with this:
    footerSumValue="BulkInvoiceSearchView.ApprovedTransactionAmount"
On the "TotalAmount" Cell, replace this:
    footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.TotalSum) :
    null"
with this:
    footerSumValue="BulkInvoiceSearchView.TotalTransactionAmount"
```

In the CheckSearchResultsLV.pcf file:

On the *Amount* Cell, replace this:

```
footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.Sum) : null"
```

with this:

```
footerSumValue="CheckSearchView.GrossAmount"
```

In the RecoverySearchResultsLV.pcf file:

On the *Amount* Cell, replace this:

```
footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.Sum) : null"
```

with this:

```
footerSumValue="RecoverySearchView.Amount"
```

Unable to access GScript from Java class (CC-52191)

Issue: The documentation is incorrect regarding generating Java types in the Java external entities (the Java Entity Libraries) for arbitrary GScript classes. The application *only* generates Java classes for types that are entities, typelists, plugin interfaces, or a type attached to any of the previously mentioned objects because of properties or methods. So, if your GScript class is not referenced by an entity, typelist, or plugin, then ClaimCenter does not generate an external Java class for it.

Workaround: To expose some arbitrary GScript class, create a property or method on an entity using a GScript enhancement, declared to return an instance of your class. If you only use static methods on the class, then your implementation can return null since the declaration alone is sufficient to include that class in the Java libraries. If no built-in entity seems appropriate for your GScript enhancement, then create a new non-persistent entity in the data model configuration files. Next, create GScript enhancements for methods or properties to return an instance of your GScript class. After you regenerate the Java Entity Libraries, your GScript class is available from Java.

RecoverySet can have at most one uncanceled recovery consistency check error (CC-57343)

Issue: Multiple voided recoveries contained within the same transaction create reports of database inconsistencies. This is because the SQL query on the consistency check is incorrect.

Workaround: You can safely ignore the consistency check.

Restored claims fail if there is no assignment and the default owner is not in a group (CC-57542)

Issue: There can be problems assigning restored claims if the default owner is not a member of a group.

Workaround: The default owner must belong to at least one group.

The ExceedsAvailableReserves method on the CheckSet entity does not work when editing a check (CC-57426)

Issue: The ExceedsAvailableReserves property on the CheckSet entity does not work correctly when you edit a check. The property verifies if the CheckSet transactions exceed the available reserves. (This also includes any offset reserves.)

Workaround: If you use this property in your rules, you must contact Guidewire Support for a possible alternative. Guidewire is aware of this issue and will address it in a future release.

InetSoft Report Scheduling is disabled in the base configuration (PL-8687)

Issue: Scheduling is disabled in the base configuration. If the scheduler is enabled for autostart by default, this interferes with restarting of the sree server if the scheduler was not shutdown before restarting sree server.

Workaround: To re-enable it, you must edit your sree.properties. Change the properties:

- `schedule.auto.down=true`
- `schedule.auto.start=true`

After the scheduler is enabled, ensure that the property `scheduler.classpath` is set correctly. Part of the build process should replace a token '@SREE_HOME@' for this property to the location where your InetSoft sree is installed on your application server. Refer to the InetSoft documentation regarding the Scheduler for more information. As of sree v.10.0, you can find it under Section 9: Scheduler of the Administration Reference.

Note the property `repository.audit.enabled` has been set to `false` overriding the default InetSoft sree value:

```
repository.audit.enabled=false
```

This disables audit records, which can cause issues with reporting if auditing is not setup properly.

Clicking the back button on a browser should show a warning popup if the MaxBrowser-HistoryItems = 0 (PL-9321)

Issue: Because Guidewire applications do not formally support or control browser Back button behavior, the Back button should not be used for navigation in Guidewire applications. However, Guidewire applications do not automatically disable a browser's Back button when they are active.

Workaround: If you want to add additional safeguards against accidental use of the Back button, do so by using standard ways to disable back buttons, including:

- Launching the application in a browser frame that has the back button removed, or
- Writing some additional Javascript to popup a dialog with text explaining why using the Back button is not a good idea, while giving the option to cancel the Back command.

In Studio, clicking Enter in a DisplayKey imbeds '\n' in the displaykey, which breaks Claim Print (PL-5560)

Issue: In Studio, selecting Enter in DisplayKey embeds a '\n' in the displaykey. Hard returns in display keys break the Claim Print functionality.

Workaround: Do not have Enter keys in your displaykeys if you want to use the print functionality.

GScript bug in anonymous inner classes implementing an inner interface (PL-9363)

Issue: GScript has a known issue in this release around implementing a Java inner interface in GScript.

Workaround: If you need to implement a Java inner interface, you have two options:

- If the interface has exactly one method, then you can take advantage of GScript blocks to implement the interface as a block. The parameters of the block are the same as the parameters to the single method, and the return type is the same as the return type of that method.
- Implement the interface in Java.

For example, the `PluginCallbackHandler` provided by `MessageReply` plugins requires you to implement `PluginCallbackHandler.Block`, an inner interface. This interface has one method, `void run()`, so you can implement it with a block that takes no arguments and has no return value.

For example, this throws an exception:

```
public function messageReceived(final messageId : int) : void {
    var myBlock : PluginCallbackHandler.Block = new PluginCallbackHandler.Block() {
        public function run() : void {
            try{
                var message = _messageFinder.findById(messageId); ===> line 111 GscriptJMSMessageReply.gs:111
                message.reportAck();
            } catch(e) {
                _logger.error("Found exception while acking " + e.printStackTrace());
            }
        }
    };
    _callbackHandler.execute(myBlock);
}
```

However, you can replace it with the following code that works:

```
public function messageReceived(messageId : int) {
    _callbackHandler.execute(\ -> {
        try {
            var message = _messageFinder.findById(messageId)
            message.reportAck()
        } catch (e) {
            _logger.error("Found exception while acking " + e.printStackTrace())
        }
    })
}
```

The InetSoft reporting software is problematic if you use the specific user-role mapping called VPM (PL-10323)

Issue: The specific user-role mapping feature in the InetSoft reporting software does not work with this version of ClaimCenter.

Workaround: Guidewire is aware of this issue and will address it in a future release. If you need additional details on how to work around this issue, contact Guidewire support.

chapter 31

Guidewire ClaimCenter 5.0.7 Release Notes

Release 5.0.7.23

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Insurance Suite, Guidewire ContactCenter, Guidewire Studio, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries. All other product names are trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

- This release of Guidewire ClaimCenter is 5.0.7.23
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 9.0, build 20090817.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter

The following issues pertain to installing this release.

Inetsoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 9.0 using the following license key:

L000-75F-ERX-00009700100001F-F2AFD44C26F6

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide*.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Software Updates

There are several software updates that were implemented since 5.0.4:

- The release supports Microsoft Internet Explorer 8
- If you use SQL Server 2005, you must install its Service Pack 3 (SP3)
- You must run 64-bit operating systems on production application servers

Archiving Updates

Archiving in ClaimCenter 5.0.6 went through major structure changes. ClaimCenter archives claims using a *Distributed Work Queue*, which archives one claim at a time, as opposed to archiving multiple claims at once. Guidewire made these changes because it found that customer configuration of extension entities and fields could very likely break the archiving process. The new methodology adds additional verification of the data model, and ensures any problems will cause archiving a claim to fail early, rather than after a claim has already been archived (and so cannot be restored).

Guidewire has also included additional checks to be run on the claim graph. You can view the results of these checks by accessing the **Internal Tools** screen. While you might not go live with archiving in this release, Guidewire recommends that you resolve these warnings before going live on either ClaimCenter 5.0.6. or 5.0.7.

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

This section describes the product changes in this release.

- Base PCF File Changes
- Base Resource Changes
- Improvements and General Issues

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.6 to 5.0.7

- 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/platform` directory, [click here](#).

Base Resource Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 5.0.6 to 5.0.7

- To view a report of the changes in the base resources in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base resources in the `modules/platform` directory, [click here](#).

Improvements and General Issues

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

ID	Description
Administration	
CLM-13448	Corrected an issue where the team statistics on the Team tab displayed the incorrect time on the non-batching server.
Archiving	
CLM-13158	Corrected an issue where you were able to archive a claim with open activities. The fix included changing the archive workitem writer so that it does not pick up claims with open activities. There is also a new rule which skips claims from archiving if there are open activities.
Core	
PL-7694	Date entries with trailing characters (such as <code>02/14/2a09</code>) are now disallowed.
PL-9944	Guidewire has modified how you access the current server environment. Do not use <code>Libraries.ServiceInfoSource.getEnv()</code> to access the current environment. Instead, use <code>gw.api.domain.ServiceInfoSource</code> .
PL-10087	Enabling Query Optimizer Tracing or Extended Query Tracing in the Web Profiler now no longer can leave database connections with tracing permanently enabled.
PL-10139	Corrected an archiving issue in which archiving a claim failed if the claim had a related contact.
PL-10373	It is now possible to include an encrypted column in an index.

ID	Description
PL-11069	Corrected an issue that caused a database upgrade error if the upgrade contained a validation rule that referred to a custom system permission. This only occurred if starting the upgrade from a new database.
PL-11074	Guidewire has disabled sorting on the <code>Value</code> column of the (Server Tools) Management Beans page.
PL-11182	Corrected an issue with database encryption in which switching from one encryption plugin to another could potentially cause the upgrade to fail because of the change in column length.
Financials	
CLM-13181	Corrected a problem with an integrity check that was previously disallowing all sharing of <code>ExchangeRate</code> entities between transactions in data to be loaded from the staging table. The integrity check now only allows sharing of exchange rates among payments on the same check.
CLM-14446	Bulk invoices with transferred placeholder checks can now be marked as <i>Issued</i> or <i>Cleared</i> through SOAP API calls without throwing exceptions in the user interface.
Gosu	
PL-10128	Fixed an issue in the GScript parser related to the <code>Deprecated</code> annotation, which could cause the server to hang. The issue was intermittent and has now been corrected.
PL-10777	Corrected an issue in which the Gosu engine evaluated the arguments to a function before evaluating the root of the function. This was inconsistent with the behavior of the Java language.
Integration	
CLM-13080	Corrected an issue where the ISO payload generation for the method <code>generateUntypedPayload()</code> failed because the description was null.
CLM-13189	Guidewire updated the ISO XSL stylesheets.
PL-9954	ClaimCenter fixed a subtle bug in how messaging destinations reads new messages from the messaging queue and dispatches them to messaging plugins. In a rare edge case, a non-claim-specific message could enter the send queue before the related claim-specific messages because of the separate database queries for these two categories of messages. Starting in this release, in each round of sending, each messaging destination always reads claim-specific messages (safe-ordered messages) from the queue before non-claim-specific messages. (Although the read order changed, the dispatch order to messaging plugins did not change. As before, messaging destination threads send non-claim-specific messages before claim-specific messages.)
PL-10116	Guidewire has modified the application behavior so that it is no longer possible for the application server and the archive server to point to the same <code>jdbcURL</code> value. This prevents the two values from accidentally pointing to the same database.
PL-10396	Guidewire has added a new configuration parameter called <code>LoadSoapServicesOnStartup</code> that controls whether the application publishes web services at application start-up (<code>true</code>) or at the time of the first request for any of the web services (<code>false</code>). The default value is <code>true</code> .
PL-10411	Corrected an issue with the <code>IMessagingToolsAPI</code> API that occurred while attempting to acknowledge messages using web services. This issue occurred if the number of field changes was bigger than the smallest number of field values in each field change.
PL-10473	Guidewire has added a <code>UseMessageCreatorAsUser</code> property to the <code>emailMessageTransport</code> plugin. By default, the <code>emailMessageTransport</code> plugin will use the system user to retrieve a document. Set this value to <code>true</code> to retrieve the document on behalf of the user who generated the email message.
PL-10548	Corrected an issue with message plugin error reporting. Whenever a message plugin threw an error, the application would store the error message in the database. If the error message description length was greater than the database column width (> 255 characters), then the database truncated the error message. (The message destination will handle the error normally.)
PL-10730	Modified the (Administration) Workflows page to show link to the parent and children of the current workflow.
PL-10970	The application server now logs an error whenever its connection status to the InetSoft report server changes.
PL-11010	Guidewire has modified the default behavior of <code>sree.properties</code> configuration parameter <code>dependency.checker.enabled</code> . The default is now <code>false</code> . This disables the InetSoft dependency finder. If you set this parameter to <code>true</code> , it can significantly impact system performance.
PL-11528	Corrected an issue with adding more than one document to a <code>ResultSet</code> if using the <code>IDocumentMetadata</code> plugin. For example, any attempt to add multiple documents to a claim on the New Reserve screen would fail.
Manageability	

ID	Description
PL-9683	Guidewire has added a new Info page to the System Tools available to administrators. Named Runtime Environment Info , it contains information on the Application Server versions, the Java Runtime version, and whenever possible, the hardware configuration.
PL-9894	Guidewire has added a (Server Tools) Server DMV Snapshot screen that displays information on the aggregate usage for all queries in the DMV, along with the percentage of resource usage by the top N queries.
PL-9910	Guidewire has modified the (Server Tools) Database Storage page to include the following information: <ul style="list-style-type: none"> • Queries Executed to Build Download • Summary of Queries Executed to Build Download
PL-9966	Guidewire has modified the (Server Tools) Database Storage page so you can choose specific tables and the mode.
PL-10049	Guidewire has corrected an issue in which Internet Explorer would become unresponsive if you clicked the Download Database Storage Info button on the Database Storage Information page in Server Tools .
PL-10109	Guidewire modified the (Server Tools) Work Queue Info page to show additional worker summary information, including throughput information.
PL-10152	Guidewire modified the (Server Tools) Info Pages, Data Distribution page and disabled the Download Zip File button if there is a database batch process running.
PL-10226	Guidewire modified the (Info Pages) Database Storage information screen to include Index Usage Statistics.
PL-10983	Guidewire modified the (Server Tools) Info Pages, Data Distribution screen to ensure that you select at least two data distributions to include in the download.
PL-11150	Guidewire added a new download to the System Tools Work Queue page. This download provides a zipped CSV file with the history of a given work queue (timing, completed items count and throughput).
Printing	
CLM-13145	Corrected an issue where there was an exception in the user interface if you printed multiple ISO pages. Now the list view printing is truncated if the <code>list</code> view is embedded and the number of rows in the <code>list</code> view exceeds the threshold defined in the <code>PrintMaxNestedListViewSize</code> parameter. The default is 100. This prevents excessive memory usage that could crash the server if these list views are too large. Non-embedded list views do not have the same memory usage issue so they are not limited.
Reporting	
CLM-414	Corrected an issue where the Loss Run report was not retrieving all claims for a selected policy.
CLM-12880	Fixed an issue where the Litigation Summary report did not group by Loss Year or Claim Group.
CLM-14458	Corrected a security issue with InetSoft reports. The fix included using username credentials instead of client information for principal creation.
CLM-14472	Fixed an issue where some ClaimCenter reports were located outside the ClaimCenter directory.
Search-Core	
CLM-14437	Corrected an issue where the Simple Claim Search ignored the <code>MaxClaimSearchResults</code> config parameter and returned all results, even if this exceeded the parameter value.
CLM-13033	Corrected an issue where you would get an error if you linked or removed and again linked a document to an activity.
CLM-13075	Fixed an issue with the Claim ACL where <code>ClaimUserRole</code> access was ignoring the claim's Access Profile. This fix also corrects an error in the security-config.xml validation process, so in rare cases where an error was reported but the server still started up, now that error will prevent the server from starting.
CLM-13147	To troubleshoot the purging of claims, Guidewire added <i>Debug</i> level logging of the graph (in DOT format) in the case where there is an invalid claim graph when purging claims. This usually occurs if you have archiving turned off, since the same validation/logging occurs at startup if archiving is turned on. Now if you go to the ArchiveInfoPage screen in the user interface, you can see any specific issues arising out of purging a claim.
CLM-13196	Corrected an issue where the claim number disappeared from the Claim tab when you viewed the Print Claim screen.
Studio	
PL-10744	Corrected an issue that occurred intermittently when making changes to PCF files. Occasionally, the application did not persist the changes after clicking Verify . Instead, the PCF file reverted to its previous state.

ID	Description
PL-10750	Guidewire modified the behavior of the Studio Entity Names editor so that it now correctly displays entities with <code>displayName</code> columns, and makes the variables used in the Entity Name GScript editor into String types for <code>displayName="true"</code> columns.
Web	
PL-10371	Corrected an issue in which setting the <code>UseArchivedStyle</code> to <code>true</code> on a Row with a <code>TextAreaCell</code> did not work correctly.
PL-11245	Corrected an issue in which the mouse cursor did not indicate that the application was busy if a <code>postOnChange</code> operation extended past three seconds. This gave the appearance that you could continue to work in the application when it was actually not possible due to background processing.

Known Issues and Limitations

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

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 login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Later versions of Internet Explorer do not allow you to set this option, and should enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928.

However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click Logout at the top of the ClaimCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer. **Warning:** Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the [HKEY_CURRENT_USER/Software/Microsoft/Windows/CurrentVersion/Explorer/BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is the default port that the ClaimCenter JMX RMI adapter uses. The ClaimCenter server will then report a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` configuration in `ClaimCenter/config/config.xml`.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands in UNIX, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

If you are using Windows, you can use Cygwin.

Periodic exception stack received on check transfer (CC-31566)

Issue: If you are using the `Claim.newExposure(subtype : ExposureType, useClaimSource : boolean)` method in your rules, and the `ExposureType` that you pass only maps to a single `LossPartyType`, then that `LossPartyType` will automatically be set into the exposure. Otherwise it will be left blank, causing an exception. This is a rare error, and is caused when one of the payments that is being transferred has an exposure with no `PrimaryCoverage` or `LossParty`.

Workaround: You should add a Pre-update rule that sets the `PrimaryCoverage` and `LossParty` fields to a non-null value.

IClaimAPI depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: `IClaimAPI` depends on `SynchStateData`, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To retrieve the payment's send date, both the Transactions and the Checks table have to be searched; this hampers performance and is not done.

Status: Guidewire is aware of this issue.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The **Reset** button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the **Reset** button an **Initialize** button.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you access the ClaimCenter **Desktop Activities** page and repeatedly click **Activities**, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to either Internet Explorer 7 or 8.

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the **Plan of Action** → **New Evaluation** page will display an incorrect title *New Evaluation* instead of *LOBName Evaluation*, and the page will not have a *LOBName Cost* field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use `ClaimEvaluationDetail` page instead of `ClaimEvaluationDetails<LOB>` page. Also, do not customize the `ClaimEvaluationDetail` page heavily.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

After upgrading from ClaimCenter 3.1, the doctor is no longer shown in medical detail exposures (CC-40595)

Issue: ClaimCenter 3.1 displayed the value `Exposure.Doctor`, but later versions of ClaimCenter display the value `Claim.FirstIntakeDoctor`. During upgrade, the data is not migrated from the old field to the new field, resulting in the upgraded ClaimCenter version showing a blank instead of the real value.

Workaround: Add another PCF field in ClaimCenter 5.0.x on the **Medical Details** page below the current **Doctor** field for `Exposure.Claim.FirstIntakeDoctor`. This displays the upgraded data.

Potential to lose LocationXXX data during upgrade (CC-41555)

Issue: Guidewire changed the location of the `LocationCity`, `LocationState`, `LocationStreet`, and `LocationZip` fields between ClaimCenter 3.1.x, 4.0.x, 5.0.0, and 5.0.1:

- In ClaimCenter 3.1.x, these fields live on the exposure.
- In ClaimCenter 4.0.x, these fields live as extensions on `MobilePropertyIncident`.
- In ClaimCenter 5.0.0, Guidewire moved these fields into a separate `LocationAddress` object, in the core data model.
- In ClaimCenter 5.0.1, Guidewire changed `LocationAddress` into an extension on `MobilePropertyIncident` (in `extensions.xml`).

Upgrading from ClaimCenter 3.1.x or 4.0.x. Guidewire added a version check that should prevent data loss when upgrading to 5.0.2 from either 4.0.x or 3.1.x. It will look for the `LocationXXX` fields in incident or expo-

sure (depending on whether you are upgrading from 4.0.x or 3.1.x). If any of those fields exist, then the version check will insist the following:

1. There is a `LocationAddress` field, of type `Address`, somewhere in the incident hierarchy
2. That any incidents/exposures with the `LocationXXX` fields set map to incident types that contain the `LocationAddress` field. That is, it checks that all `LocationXXX` fields that contain a value have a `LocationAddress` field to use in migrating the data.

If either of these checks fail, the version check will abort the upgrade and you must then follow the steps outlined in the workaround to solve the problem.

Upgrading from ClaimCenter 5.0.0. If you are upgrading from ClaimCenter 5.0.0, there is already a `LocationAddress` field. However, in 5.0.0, Guidewire placed the field in the core data model, while in 5.0.1 and later it is in `extensions.xml`. So, if your target version `extensions.xml` does not contain a `LocationAddress` field, then the `LocationAddress` field could get dropped entirely during the upgrade process, possibly losing data. This is the case that the upgrade version trigger does not address.

Upgrading from ClaimCenter 5.0.1 or later. This is not an issue.

Workaround: Depending on the release from which you are upgrading, you need to do the following:

- **ClaimCenter 3.1.x base release.** In 3.1.x, the `LocationXXX` fields were in the base data model, on exposure. As part of the upgrade, you will need to decide which exposure types map to which incident subtype. You must map any exposure types that have the `LocationXXX` fields set to an incident subtype that has the `LocationAddress` field. This ensures that the upgrade process has a place to move the data.
- **ClaimCenter 4.0.x base release.** In 4.0.x, the `LocationXXX` fields were in `extensions.xml`, in the incident hierarchy. You could choose where in the incident hierarchy to put these fields. In upgrading to any 5.0.x version, replace these fields in `extensions.xml` with `LocationAddress`, at the same level in the hierarchy.
- **ClaimCenter 5.0.0 base release.** In 5.0.0, the `LocationAddress` field was in the base data model, in `MobilePropertyIncident`. If you are upgrading to 5.0.x, you will need to make sure `LocationAddress` is in the `extensions.xml` file. Guidewire recommends that you put it in `MobilePropertyIncident`. However, if desired, it is possible—because it now lives in `extensions.xml`—to move it to a different place in the incident hierarchy.

Transaction `runWithNewBundle` Does Not Follow Links (PL-1716)

Issue: The GScript API that runs a block of code with new bundle (`gw.transaction.Transaction.runWithNewBundle(...)`) has some limitations. Even if you call `newBundle.add(yourentity)`, the `add` method does not follow the graph links to other entities to which it links. You must manually add each sub-object to the new bundle, using the call `bundle.add(linkedEntity)` for each entity. This includes all foreign key references to other entities as defined in the built-in data model files, but also includes customer data model extensions that include foreign key references.

Workaround: You can work around this by committing all entities in the current bundle instead of using this API. This may be an issue when designing implementing a custom web service. Committing all entities in the current bundle includes all entities serialized into the bundle as part of the web service request because they were parameters to a web service method. There may be cases where entities that are web service API parameters you do **not** want commit to the database.

One workaround is to redesign the web service API so instead of taking entities (the entities you do not want to commit) it takes GScript class instances instead of Guidewire entities. Your custom GScript classes can provide only the fields that you want to pass in your integration code, which may be a much smaller subset of fields in some cases. Because they are GScript classes and not entities, committing the entire bundle with `gw.transaction.Transaction.getCurrent().commit()` will skip that data that you choose not to commit to the database.

Creating a new document from a template results in a `ClassNotFoundException` (CC-42867)

Issue: If you implement the `IDocumentProduction` plugin in Java, then when you create a new document from a template you may encounter a `ClassNotFoundException`.

Workaround: Copy the file `ClaimCenter/modules/cc/config/plugin/ExternalMappings.properties` to `ClaimCenter/modules/configuration/config/plugin`. Then add the following line to the copied file:

```
gw.api.document.DocumentCreationInfo = gw.api.document.DocumentCreationInfo
```

In addition, copy the file `ClaimCenter/modules/cc/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf` to `ClaimCenter/modules/configuration/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf`. In the copied file, replace the following line:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.DocumentTemplateDescriptor
      .TemplateId, gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

with the following:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    util.document.DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.
      DocumentTemplateDescriptor.TemplateId,
      gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

After upgrade, two duplicate incidents are created (CC-44646)

Issue: If you have two exposures with the same incident type details (for example, auto damage on the same car with the same driver), the database upgrader creates two identical incidents.

Status: Guidewire is aware of this issue.

Limitation in reporting when a typecode name contains a comma (CC-45305)

Issue: Due to the limitations of the InetSoft reporting software, issues arise when a passing a typelist as a parameter to a drill-down report if one of its typecode names contains a comma. (For example, typelist Loss Cause contains a typecode with the name *Fall, slip, or trip injury*.) If a typecode name does contain a comma, InetSoft thinks it is an array and sends multiple values. Guidewire has implemented a solution for the Loss Cause typelist in ClaimCenter. However, if you are passing any other typelist as a parameter to a drill-down report and it contains a typecode with a comma, you will experience this issue.

Workaround: Rename any commas from the Name attribute of the typecode.

Typelist error message: "contains a typecode with an empty code/name" (CC-46085)

Issue: When starting the server, you may see an error that says a typelist contains a typecode with an empty code or name. This error is mistakenly generated and does not indicate any problem.

Workaround: Ignore this error.

Upgrade fails when upgrading CoverageType typelist that already has a code of: BLDG (CC-46510)

Issue: During upgrade, you receive an exception for the following upgrade trigger: `RecodeCoverageTypecodesTriggers$RecodeCoverageTypecode` and your Coverage Type typelist already has a typekey with code =BLDG

Status: Guidewire is aware of this issue.

Unable to subclass `gw.api.quickjump.SpecificClaimCommand` (CC-49433)

Issue: If you extend the `gw.api.quickjump.SpecificClaimCommand` class to override the `isPermitted()` method, then the Specific Claim QuickJump command is unavailable. It also can cause an error during server startup.

Workaround: You can create a new QuickJump command that subclasses `com.guidewire.pl.web.navigator.commands.DefaultQuickjumpCommand`, and which delegates its operations to an instance of the `gw.api.quickjump.SpecificClaimCommand`. . If you need to create a `SpecificClaimCommand`, contact Guidewire Support for sample code attached to issue PL-4061.

Iterator buttons not working correctly in `MatterDetailsDV.pcf` (CC-48879)

Issue: On the `MatterDetailsDV` file, the **Add** button does not hide when not in *Edit* mode. Also, the **Add** and **Remove** buttons do not show when in *Edit* mode, on the `StatusLinesLV` file.

Workaround: Set `lockWhileEditing="false"`. Note it is not locked.

`ShowNewExposureChooseByCoverageMenuForLossTypes` parameter does not allow a blank value (CC-49426)

Issue: If the `ShowNewExposureChooseByCoverageMenuForLossTypes` configuration parameter is not defined in the `config.xml` file, then the server throws an exception trying to look for at least one loss type in this parameter. This is problematic if you do not want the menu to be displayed.

Workaround: Remove the menu from the PCF files.

FNOL wizard not saving selected vehicle (CC-47439)

Issue: The second step of the FNOL wizard does not save the selected vehicle when viewing the **Policy Details** screen and 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)"`.

Integrity checks allow offsetting payments to be associated to a different check than the payment it offsets (CC-49866)

Issue: Integrity checks for staging tables do not ensure that the two payments pointed to by a `TransactionOffsetOnset` row belong to the same check.

Workaround: You must correct the errors in data mapping, and ensure that each Voided or Stopped payment on a Voided or Stopped check also has an offset payment (with negative amount, in *Submitted* status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

Exception stack trace in user interface when `DisplayableException` thrown from Policy plugin (CC-47987)

Issue: A `GScript` util class throws a `DisplayableException` to the policy plugin `GScript` class which is again throwing another `DisplayableException` with the message from the first exception. The exception message is not user friendly.

Workaround: Add the following parameter to your `config.xml` file:

```
<param name="IntegratedStackTraces" value="true"/>
```

ArithmeticException when exchange rate is set to 0 (CC-48438)

Issue: When you enter an exchange rate = 0 in the New Check wizard, you see an `ArithmeticException` error in the user interface.

Workaround: Add a validation expression (attribute `validationExpression` or `requestValidationExpression`) to the `Transaction_ExchangeRate` element in `ExchangeRateInputSet.default.pcf` and `ExchangeRateInputSet.Check.pcf` files.

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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 info, 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.

Error when removing claim flags through the Team tab (CC-50159)

Issue: The code to remove the flagged status from a claim in the Team view only works with one claim at a time. The following PCFs allow you to select multiple claims in the LV and then click **Remove Flag**, which results in an error. This can happen with these PCFs: `TeamGroupOtherClaims.pcf`, `TeamGroupPendingClaims.pcf`, `TeamUserClaims.pcf`

Workaround: Modify the `flags` attribute of the Remove Flag button in each PCF to read:

```
flags="one CanRemoveFlag"
```

Group cache needs correct sizing (CC-51545)

Issue: Guidewire discovered assignment performance issues during the ClaimCenter 5.0.5 performance testing on a 2.5M claim database due to the fact that the default configuration of the `GroupCache` was not large enough to hold all the groups. This resulted in frequent premature evictions from the cache during assignment and subsequent thrashing of the cache.

Workaround: Increasing the cache size to make it large enough to hold all the groups (2200 in this case) significantly helps assignment performance. If possible, Guidewire recommends that you configure the value for the GroupCache size (located in the `config.xml` file) to make it large enough to hold all groups.

When editing a check, check portions should not exceed total payments (CC-45520)

Issue: In the Edit Check wizard, the fixed-amount check portions are not prevented from exceeding the total of the payments on the multi-payee check.

Workaround: You must make the following configuration change in the `EditCheckWizard.pcf` file: add the method `Wizard.validatePayments` to the `onExit` expression for the `NewCheckPayments` wizard step.

Problematic calls to SOAP API methods (CC-45378)

Issue: Calling ClaimCenter SOAP API methods from within a messaging plugin can be problematic because the root entity is locked during message processing.

Workaround: If you are upgrading to 5.0.x and your 4.0.x or earlier configuration is calling SOAP API functions from within messaging plugins (or message sinks), you need another approach. Contact Guidewire Support for information on alternatives to calling SOAP APIs from within your messaging integration code.

Invalid activity pattern referenced in a workflow prevents ClaimCenter from starting (PL-5987)

Issue: If a workflow configuration file has an `ActivityStep` that references an activity pattern that does not yet exist in the system or database, (the activity pattern could be created after deployment in a real production release), then the application fails to start.

Workaround: Do the following:

- Remove all the references to the workflow activity pattern
- Deploy
- Create the activity pattern
- Re-deploy the application with the references to the workflow activity pattern in place.

Duplicate descriptions for image/jpg mime types in config.xml (CC-50371)

Issue: In the `config.xml` file, there are two entries under MIME types for jpeg: one for pjpeg and one for normal jpeg. This causes a problem as the descriptions of both are identical. If you upload a normal 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 Studio.

License state drop down menu in the user interface produces unexpected results (CC-50529)

Issue: If you use the `AddressAutocompleteUtil.getStates()` method to populate license states in the `FNOLVehicleIncidentPopup.pcf` file, then you might see results that are not expected, such as countries or retired typecodes. This occurs when you select the license state drop down menu in the user interface. What is happening is that `AddressAutocompleteUtil.getStates` is passing true to the `getTypeKeys` method when it should be false.

Workaround: Use the following:

```
function getStates_Ext(country : typekey.Country) : typekey.State[] {
    var statelist = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = "US"; //default to US
    }
}
```

```

    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
            stateList.add(eachState)
        }
    }
    return stateList.toArray() as typekey.State[]
}

```

Include this function in your .pcf file. Instead of calling the AddressAutocompleteUtil method in the valueRange, call this method instead:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

Incorrect additivity statements in the default logging.properties file (CC-50615)

Issue: Some additivity statements in the logging.properties file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the logging.properties file (located at modules/cc/config/logging), change:

```
log4j.additivity.PluginsLog=false
```

to

```
log4j.additivity.Plugin=false
```

The statements for API, Database, Messaging, Messaging.ISO, Plugin.ClaimNumGenLog, Plugin.IPolicySearchAdapter, and Plugin.IContactSearchAdapter are also incorrect and need to be changed.

Claim association can result in data base consistency check failure (CC-50724)

Issue: Making a claim association from the user interface can result in making two claims as primary. This results in data base consistency check failure.

Workaround: In the user interface, edit associations and make *only one* claim as primary.

Paging controls are disabled while trying to reassign an activity (CC-50899)

Issue: In the user interface, the paging controls become disabled when you try to reassign an activity by searching for a queue.

Workaround: In Studio, set startInEditMode=true in the AssigneePickerPopup. This is located in the **Advanced Properties** section.

Duplicate check search velocity template is incorrect (CC-50688)

Issue: There is a bug in the Velocity template used to construct the query for finding duplicate checks in the check wizard. It does not handle all the cases where one of the ServicePdStart/ServicePdEnd fields is null on one check but not on the other. Usually this does not matter because both these fields are set together so they are both either set or not set, in which case you can use the existing template.

Workaround: Find the following lines in the check.vm file in Studio under configuration → Other Resources → duplicate-search. The template can be changed to be more consistent. Instead of:

```

#if ($Check.servicePdStart)
AND cc_check_head.ServicePdEnd > $Util.sql.getSQLStringValue($Check.servicePdStart)
#end
#if ($Check.servicePdEnd)
AND cc_check_head.ServicePdStart < $Util.sql.getSQLStringValue($Check.servicePdEnd)
#end

```

The template can be changed to:

```

#if ($Check.servicePdStart && $Check.ServicePdEnd)
AND cc_check_head.ServicePdEnd >= $Util.sql.getSQLStringValue($Check.servicePdStart)
AND cc_check_head.ServicePdStart <= $Util.sql.getSQLStringValue($Check.servicePdEnd)
#end

```

```
#if (!$Check.servicePdStart || !$Check.ServicePdEnd)
AND (cc_check_head.ServicePdStart IS NULL OR cc_check_head.ServicePdEnd IS NULL)
#end
```

The adjusted version compares check service periods if both start and end values are set. Otherwise, it assumes all service periods with either only one bound or neither bounds set match. So checks with complete service periods (both lower and upper bound set) match if the service periods overlap. Checks with incomplete service periods all match (at least as far as their service periods are concerned).

Since this template is configurable, you should not take these changes if you have already customized this logic to your business requirements. Also, you do not need to make these changes if you are satisfied with the current matching behavior.

Checks Search and Recoveries Search screens execute a query to populate the footer even when no selection is made (CLM-7750)

Issue: On single-currency configurations, this change makes a workaround possible for the problem of the sum query being unnecessarily evaluated on the search screens.

Note: This workaround will not produce correct results for servers running in multi-currency mode.

Workaround: To implement this workaround, you must make the following .pcf file changes in Studio:

In the BulkInvoiceSearchResultsLV.pcf file:

```
On the "ApprovedAmount" Cell, replace this:
    footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.ApprovedSum)
    : null"
with this:
    footerSumValue="BulkInvoiceSearchView.ApprovedTransactionAmount"
On the "TotalAmount" Cell, replace this:
    footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.TotalSum) :
    null"
with this:
    footerSumValue="BulkInvoiceSearchView.TotalTransactionAmount"
```

In the CheckSearchResultsLV.pcf file:

On the *Amount* Cell, replace this:

```
footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.Sum) : null"
```

with this:

```
footerSumValue="CheckSearchView.GrossAmount"
```

In the RecoverySearchResultsLV.pcf file:

On the *Amount* Cell, replace this:

```
footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.Sum) : null"
```

with this:

```
footerSumValue="RecoverySearchView.Amount"
```

RecoverySet can have at most one uncanceled recovery consistency check error (CLM-512)

Issue: Multiple voided recoveries contained within the same transaction create reports of database inconsistencies. This is because the SQL query on the consistency check is incorrect.

Workaround: You can safely ignore the consistency check.

Restored claims fail if there is no assignment and the default owner is not in a group (CC-57542)

Issue: There can be problems assigning restored claims if the defaultowner is not a member of a group.

Workaround: The default owner must belong to at least one group.

The ExceedsAvailableReserves method on the CheckSet entity does not work when editing a check (CC-57426)

Issue: The ExceedsAvailableReserves property on the CheckSet entity does not work correctly when you edit a check. For this issue to be problematic, you must have explicitly used CheckSet.ExceedsAvailableReserve in rules, which is only useful when the configuration parameter AllowPaymentsExceedReservesLimits = true. With the default setting of false, the Check wizard ensures that a check's payments do not exceed available reserves, including in edit mode.

Status: Guidewire is aware of this issue.

InetSoft Report Scheduling is disabled in the base configuration (PL-8687)

Issue: Scheduling is disabled in the base configuration. If the scheduler is enabled for autostart by default, this interferes with restarting of the sree server if the scheduler was not shutdown before restarting sree server.

Workaround: To re-enable it, you must edit your sree.properties. Change the properties:

- schedule.auto.down=true
- schedule.auto.start=true

After the scheduler is enabled, ensure that the property scheduler.classpath is set correctly. Part of the build process should replace a token '@SREE_HOME@' for this properly to the location where your InetSoft sree is installed on your application server. Refer to the InetSoft documentation regarding the Scheduler for more information. As of sree v.10.0, you can find it under Section 9: Scheduler of the Administration Reference.

Note the property repository.audit.enabled has been set to false overriding the default InetSoft sree value:

```
repository.audit.enabled=false
```

This disables audit records, which can cause issues with reporting if auditing is not setup properly.

Clicking the back button on a browser should show a warning popup if the MaxBrowser-HistoryItems = 0 (PL-9321)

Issue: Because Guidewire applications do not formally support or control browser Back button behavior, the Back button should not be used for navigation in Guidewire applications. However, Guidewire applications do not automatically disable a browser's Back button when they are active.

Workaround: If you want to add additional safeguards against accidental use of the Back button, do so by using standard ways to disable back buttons, including:

- Launching the application in a browser frame that has the back button removed, or
- Writing some additional Javascript to popup a dialog with text explaining why using the Back button is not a good idea, while giving the option to cancel the Back command.

In Studio, clicking Enter in a DisplayKey imbeds '\n' in the displaykey, which breaks Claim Print (PL-5560)

Issue: In Studio, selecting Enter in DisplayKey embeds a '\n' in the displaykey. Hard returns in display keys break the Claim Print functionality.

Workaround: Do not have Enter keys in your displaykeys if you want to use the print functionality.

GScript bug in anonymous inner classes implementing an inner interface (PL-9363)

Issue: GScript has a known issue in this release around implementing a Java inner interface in GScript.

Workaround: If you need to implement a Java inner interface, you have two options:

- If the interface has exactly one method, then you can take advantage of GScript blocks to implement the interface as a block. The parameters of the block are the same as the parameters to the single method, and the return type is the same as the return type of that method.
- Implement the interface in Java.

For example, the `PluginCallbackHandler` provided by `MessageReply` plugins requires you to implement `PluginCallbackHandler.Block`, an inner interface. This interface has one method, `void run()`, so you can implement it with a block that takes no arguments and has no return value.

For example, this throws an exception:

```
public function messageReceived(final messageId : int) : void {
    var myBlock : PluginCallbackHandler.Block = new PluginCallbackHandler.Block() {
        public function run() : void {
            try{
                var message = _messageFinder.findById(messageId); ==> line 111 GscriptJMSMessageReply.gs:111
                message.reportAck();
            } catch(e ) {
                _logger.error("Found exception while acking " + e.printStackTrace());
            }
        }
    };
    _callbackHandler.execute(myBlock);
}
```

However, you can replace it with the following code that works:

```
public function messageReceived(messageId : int) {
    _callbackHandler.execute(\ -> {
        try {
            var message = _messageFinder.findById(messageId)
            message.reportAck()
        } catch (e) {
            _logger.error("Found exception while acking " + e.printStackTrace())
        }
    })
}
```

The InetSoft reporting software is problematic if you use the specific user-role mapping called VPM (PL-10323)

Issue: The specific user-role mapping feature in the InetSoft reporting software does not work with this version of ClaimCenter.

Status: Guidewire is aware of this issue.

Deleting a user who owned an archive claim and running the archive worker item throws an exception (CLM-12949)

Issue: If a user is deleted after a claim has been archived and restored, then the archive worker item throws a `DBDuplicateKeyException` exception.

Workaround: Set the user who owns an archive claim to *inactive* instead of deleting them.

Email plugin has intermittent transmission failures with incorrect address message (PL-10549)

Issue: The Email plugin intermittently fails to send emails and reports that the message address is invalid, even if the address is actually valid. This intermittent problem has been reported in a small number of cases.

Workaround: Restarting the application server usually resolves the message address problem. If this is a recurring issue, then Guidewire recommends that you implement your own email message transporter. This message transporter must have the ability to read the fields from the email object to retrieve the contents of the email.

AuthoringDate field is null if ClaimCenter creates a Claim Flag Change note. (CLM-11002)

Issue: If you create a note with a subject of Claim Flag Change, the note's date always appears to be the most recent even if that is not true. This is because there is no authoring date.

Workaround: Use the following code to correct this:

```
uses java.util.Date;

for(n in claim.Notes)
  if(n.New){
    n.setFieldValue( "AuthoringDate", new Date() )
  }
}
```

An exception occurs if you add a second attachment to an email (CLM-14601)

Issue: You cannot add a second attachment to an email. This only occurs if the IDocumentMetadata plugin is enabled.

Workaround: If you need to add another attachment, then you must create another email and add the second attachment to it.

Clicking the Edit button for a policy when the Policy entity is configured with InternalOnly fields brings up prompt instead of a confirm dialog box (CLM-13224)

Issue: If you click the Edit button for a policy when the Policy entity is configured with InternalOnly fields, the user interface displays a prompt instead of a confirm dialog box.

Workaround: Make the following edits to the ClaimPolicyGeneral.pcf file.

Remove the button with ID ClaimPolicyGeneral_EditButton3 and replace it with the following (substituting proper display keys for the hard coded labels):

```
<ToolBarButton
  hideIfEditable="true"
  id="ClaimPolicyGeneral_EditButton5"
  label="displaykey.Button.Edit"
  visible="Claim.Policy.Verified and HasInternalOnlyFields and
(perm.Policy.makeeditable(Claim) and perm.Policy.edit(Claim))">
  <MenuItem
    action="gw.api.policy.ClaimPolicyMakeEditableUtil.makePolicyEditable(CurrentLocation,
Claim, true)"
    confirmMessage="displaykey.JSP.ClaimPolicyGeneral.EditEntireVerifiedPolicy"
    hideIfEditable="true"
    id="EditFullPolicy"
    label=""Edit Full Policy""
    visible="perm.Policy.makeeditable(Claim)"/>
  <MenuItem
    action="CurrentLocation.startEditing()"
    hideIfEditable="true"
    id="EditInternalOnly"
    label=""Edit Internal Only""
    visible="perm.Policy.edit(Claim)"/>
</ToolBarButton>
```

Some integrity checks do not report LUWID for bad rows (CLM-13028)

Issue: Some of the integrity checks called during staging table loading do not report the LUWID of bad rows,. It is reported as null. This prevents the automated loading program that you may write from excluding bad data from each load batch, making loading more difficult.

Workaround: Perform one of the following:

- Do incremental Integrity Checking after inserting each claim to the staging tables. After inserting the data for each claim, run `ITableImportAPI.integrityCheckStagingTableContents()`. If there is an error, you know that the latest claim/LUWID is bad, and you can insert a `cc_loadexclusion` row for it. Integrity checks are

faster than the phase of actual loading the staging data into the production tables, so this should be faster than loading each claim one at a time.

- **Optimistic import:** Insert all claims in the batch, and run `IntegrityCheckAndLoad`. If there is an error, back off and insert them one at a time. Clear the staging tables and start the batch over, according to the previous bullet. If most of your data is clean, you might consider this approach for efficiency.

User is allowed to delete a queue that has open activities (CLM-14576)

Issue: If you delete a queue that has activities assigned to it, the activities cannot be reassigned to a user. The Group Admin was allowed to delete a queue that had an open activity. Since the queue is now retired, the activity is now visible only from the **Workplan** screen. The activity can no longer be taken from the queue, since the queue is no longer visible. This means that the claim can never be closed since it has an open activity that cannot be skipped or completed.

Workaround: To prevent accidental deletion of a queue to which one or more open activities are assigned, use a `ToolBarFlag` to control availability of the **Remove** button in **GroupQueues** detail view. The **Remove** button should be available *if and only if* the user has selected one or more queues that have no open activities.

CheckRpt should load transaction currency instead of reporting currency (CLM-14586)

Issue: If you use the staging table load to import a check with a non-default (foreign) currency, you will get the incorrect currency in the `ClaimRpt` table if you use the `allreferencesallowed` option. This results in an incorrect currency for the check amount seen in the **Financials** → **Checks** page.

Workaround: Run the table import without the `allreferencesallowed` option. This limitation may affect your data conversion strategy.

Error retrieving replets from report server when you sync from the Report Admin (PL-11913)

Issue: If you click the **Sync** button for the first time in the Report Admin (**Administration** → **Report Admin** → **Reports**) you may see an error when retrieving replets from the report server.

Workaround: You must modify the `InetSoft web.xml` to set `load-on-startup` to `true` for the `ServletRepository` with the following:

```
<servlet>
  <servlet-name>replets</servlet-name>
  <servlet-class>inetsoft.sree.web.ServletRepository</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Deleting a report that is used by a permission set creates an exception in the user interface (PL-12009)

Issue: If you delete a report that is used by a permission set in ClaimCenter, then you will see a Stack Trace error in the **Permission Sets** tab on the permission set that included the deleted report. (The path is **Administration** tab → **Report Admin** → **Permission Sets**.)

Workaround: Remove the report from the permission set before deleting it from InetSoft and the ClaimCenter **Reports** tab.

Report is still visible in the Report Admin tree after you sync with the Report server (PL-12010)

Issue: If you set a report's visibility to `false` in InetSoft, you can still see the report in the ClaimCenter Admin Report Tree. The path is **Administration** tab → **Report Admin**. However, these reports still remain hidden to any report users from both the InetSoft Portal and in ClaimCenter.

Status: Guidewire is aware of this issue.

chapter 32

Guidewire ClaimCenter 5.0.8 Release Notes

Release 5.0.8.9

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Insurance Suite, Guidewire ContactCenter, Guidewire Studio, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries. All other product names are trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 5.0.8.9.

This release of Guidewire ClaimCenter supports InetSoft StyleReport Enterprise Edition 9.0, SREE v9.0 20101026 for Standard Reporting.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter

The following issues pertain to installing this release.

Inetsoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 9.0 using the following license key:

L000-74B-ERX-000097001000006-F2B01CA3EE0C

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide*.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Software Updates

There are several software updates that were implemented since 5.0.4:

- The release supports Microsoft Internet Explorer 8
- If you use SQL Server 2005, you must install its Service Pack 3 (SP3)
- You must run 64-bit operating systems on production application servers

Archiving Updates

Archiving in ClaimCenter 5.0.6 went through major structure changes. ClaimCenter archives claims using a *Distributed Work Queue*, which archives one claim at a time, as opposed to archiving multiple claims at once. Guidewire made these changes because it found that customer configuration of extension entities and fields could very likely break the archiving process. The new methodology adds additional verification of the data model, and ensures any problems will cause archiving a claim to fail early, rather than after a claim has already been archived (and so cannot be restored).

Guidewire has also included additional checks to be run on the claim graph. You can view the results of these checks by accessing the **Internal Tools** screen. While you might not go live with archiving in this release, Guidewire recommends that you resolve these warnings before going live on any release higher than ClaimCenter 5.0.6.

Changes in this Release

For a description of the feature changes between ClaimCenter 4 and ClaimCenter 5, refer to “What’s New and Changed in ClaimCenter 5.0.0” in the *Upgrade Guide*.

This section describes the product changes in this release.

- Base PCF File Changes
- Base Resource Changes
- Improvements and General Issues

Base PCF File Changes

ClaimCenter release 5.0.7 to 5.0.8

There were no changes to the base PCF files.

Base Resource Changes

ClaimCenter release 5.0.7 to 5.0.8

There were no changes to the base resource files.

Improvements and General Issues

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

ID	Description
Assignment, Email, Notes, Documents	
CLM-14551	Fixed an issue that prevented an administrative user from deselecting a document template attached to an Activity Pattern. This fix allows the user to select none appropriately.
Claim - Summary, Loss Details, Exposures, Incidents, Litigation, Etc.	
CLM-15797	Changes to the ACL were causing updates to the Claim entity, resulting in unexpected ConcurrentDataChangeException exception. Claim ownership of the Claim.Access array has been removed so that changes to the array no longer trigger Claim updates.
Contacts, Roles	
CLM-14608	The ExceptionConstraint in entityroleconstraints-config.xml was not implemented correctly, allowing Contacts of any type if an ExceptionConstraint was included. The ExceptionConstraint is now working as designed, allowing only Contacts of the type designated in the main constraint or the exception constraints.
Documentation	
CLM-14853	Made minor changes to the financials object status APIs. <ol style="list-style-type: none"> 1. The IClaimFinancials web service interface has four new methods for voiding and stopping checks: <ul style="list-style-type: none"> • stopCheck • voidCheck • stopAndReissueCheck • voidAndReissueCheck 2. There are four new supported check status transitions: from either status Voided or Stopped to the status Issued or Cleared.
Financials	

ID	Description
CLM-14732	Fixed an issue in which if you re-issued a check, the new CheckPortion entity was created using amounts from the original check's CheckRpt entity amount. This lead to inconsistencies with the original check's amount and new check's amount because the check amount and the CheckRpt amount calculation were different.
CLM-15137	Fixed an issue during denial of a check or payment. If the check had a final payment that had closed it's exposure or claim, but the exposure or claim was now open at the time of denial, an error would occur if ClaimCenter tried to reopen the claim or exposure. The deny method now checks if the exposure or claim is already open before attempting to reopen it.
CLM-15336	Corrected an issue on the Check Transfer screen related to adding new required fields to the page. If the first attempt to transfer the check had missing required fields, subsequent attempts to fix the problem and complete the operation would still display an error. The Transfer button now properly rolls back its work if screen validation fails, so another attempt to transfer will be successful.
CLM-15360	Added a more meaningful error message if a check delete fails. Previously an extra error displayed if you were trying to delete the check and you did not have permission. Previously, the error's text was "{0}" or "null".
CLM-14579	Updated ReserveLine filtering to allow pending approval transactions to be displayed.
CLM-13010	Fixed an issue in which calling the BulkInvoiceItem.setNonEroding method more than once before a commit would not work and could leave T-accounts in an incorrect state.
CLM-13098	Fixed an issue in which users saw an error message if they deleted a denied check with offset reserves.
Performance	
CLM-15735	Replaced OR with UNION in a poor-performing query to improve ClaimException performance.
Platform	
PL-10323	Modified the way ClaimCenter supports InetSoft VPM. This work has tightened the Guidewire SOAP security for ISREEAuthenticationAPI and added three more properties to sree.properties. <ul style="list-style-type: none"> gw.soapuser.username: The user name used by InetSoft to retrieve user/group/role information used by VPM and authentication gw.soapuser.userpassword: The password associated with the user name. gw.usercache.size: The cache size for user cache, group cache, and role cache.
PL-11600	Removed xe1em.jar (used for parsing Excel XML) from the application. Guidewire now provides this functionality through other means.
PL-11719	Added support for Microsoft Windows 7 as an operating system for application clients.
PL-12463	Fixed an issue that caused an 'Attempt to access bean of type "GroupUser" with a null bundle' error when retrying the assignGroupDynamically and the assignUserDynamically assignment methods after a Validation warning.
PL-12648	Corrected an issue that caused a report to freeze if that report contained freehand tables along with large data sets. This issue was due to a bug in the InetSoft v9 reporting software.
PL-12872	Improved Studio activation performance if using SCM systems that maintain a local repository (for example, SVN and CVS) and Studio is configured with proper repository mask in Tools -> Options.
PL-13687	Modified Studio to provide ability for time-sensitive diagnostic logging in the Studio console. You can enable this functionality by creating a studio.properties file in Other Resources → logging and populating it with the following text: <pre># enable studio debug level logging log4j.category.Studio=DEBUG</pre>
PL-13798	Fixed an issue that cause a CDCE exception (ConcurrentDataChangeException) due to stale cache data.
PL-13882	Fixed an issue in which a Claim search using the "Any Party Involved" option did not return the correct response if the value entered for Organization Name could match a Place name.
PL-14417	Corrected a regression issue in which the application failed to fill in the Description, Type, and MIME Type fields if you created a new document from a template.
Reporting	
CLM-12909	Fixed an issue in which the InetSoft report calculation for Catastrophe Financials: Average Loss Amount was different than the current Catastrophe Financials report. Now the Average Loss Amount is the same for both reports.

ID	Description
CLM-12957	Fixed an issue that caused some claims to be non-reported if drilling down from Open Claim Financials report to the Transaction Detail report.
CLM-12971	Fixed an issue in which claim level transactions were not included in Catastrophe Financials reports. The fix included: <ul style="list-style-type: none"> • Adding a new physical view and logical model linking the Claim entity to the Transaction entity (ccpvClaimTransaction, cclmClaimTransaction) • Repointing the table to the new logical model cclmClaimTransaction. This change now captures claim level financials.
CLM-12972	Previously, the count columns in the five Financials → Monthly reports listed the count of transaction sets that had the respective transaction type and cost type (for example, claim cost payment). The averages also used this count. Now the count columns reflect the number of claims that contained a transaction set with the respective transaction type and cost type. The Average columns also use the claim count.
CLM-13012	Modified the Report → Financials → Monthly Expense Paid report to make the Approval Month range a required selection.
CLM-13050	Fixed an issue in reporting where the Transactions Detail report only displayed Payment transactions, but did not show Recovery, RecoveryReserve, and Reserve transactions.
CLM-13051	Corrected multiple issues with the Monthly Recoveries report: <ul style="list-style-type: none"> • The recovery count was incorrect. • The drill-down to the Transaction Detail report did not show the recovery transactions. • The query parameters did not match the results.
CLM-13113	Modified the SIU reports (Payees with Multiple Claims and Multiple Roled Payees) so that drilling down through the claim number opens the Claim Summary page.
CLM-13114	Fixed an issue that caused the Report → Financials → Monthly Recoveries report to display an incorrect value for Recovery Average .
CLM-13127	Modified the Reopened Claims List report to display claim-level financials.
CLM-13754	New Document objects created from a template now correctly copies fields from the template.
CLM-14768	Modified the way that ClaimCenter calculates data in the following report files: <ul style="list-style-type: none"> • ClaimCenter\Activity List drill down.srt • ClaimCenter\Transaction Detail Group Dashboard.srt • ClaimCenter\Transaction Detail Group.srt • ClaimCenter\Transaction Detail LOB.srt • ClaimCenter\Transaction Detail Loss Type.srt • ClaimCenter\Transaction Detail.srt • ClaimCenter\Claims\Current Overdue Activities.srt • ClaimCenter\Claims\Overdue Activities.srt • ClaimCenter\Dashboards\Past 30 Day Financials.srt • ClaimCenter\Financials\Average Claim Cost Trend.srt • ClaimCenter\Financials\Average Expenses Trend.srt • ClaimCenter\Financials\Transaction Detail.srt
CLM-14670	Fixed an issue that caused out-of-memory errors.

Known Issues and Limitations

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

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.

Checks Search and Recoveries Search screens execute a query to populate the footer even when no selection is made (CLM-7750)

Issue: On single-currency configurations, this change makes a workaround possible for the problem of the sum query being unnecessarily evaluated on the search screens.

Note: This workaround will not produce correct results for servers running in multi-currency mode.

Workaround: To implement this workaround, you must make the following PCF file changes in Studio:

In the BulkInvoiceSearchResultsLV.pcf file:

```
On the "ApprovedAmount" Cell, replace this:
    footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.ApprovedSum)
    : null"
with this:
    footerSumValue="BulkInvoiceSearchView.ApprovedTransactionAmount"
On the "TotalAmount" Cell, replace this:
    footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.TotalSum) :
    null"
with this:
    footerSumValue="BulkInvoiceSearchView.TotalTransactionAmount"
```

In the CheckSearchResultsLV.pcf file:

On the *Amount* Cell, replace this:

```
footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.Sum) : null"
```

with this:

```
footerSumValue="CheckSearchView.GrossAmount"
```

In the RecoverySearchResultsLV.pcf file:

On the *Amount* Cell, replace this:

```
footerLabel="(criteria.ShowSumRow) ? gw.api.util.CurrencyUtil.renderAsCurrency(criteria.Sum) : null"
```

with this:

```
footerSumValue="RecoverySearchView.Amount"
```

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands in UNIX, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

If you are using Windows, you can use Cygwin.

IClaimAPI depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: IClaimAPI depends on SynchStateData, which in turn uses deprecated functions such as `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

In the Transactions custom/all types list view, the 'Date' column does not always reflect the scheduled send date for a recurring check's future payment (CC-34473)

Issue: To retrieve the payment's send date, ClaimCenter would have to search both the Transaction and the Check tables. This effort would hamper performance and is not done.

Workaround: Guidewire is aware of this issue.

Reset button does not reset the address fields part of proximity search (CC-35764)

Issue: On the assignment search screen, the proximity search center is pre-populated. The Reset button does not clear this field, but instead returns it to the initial value (the claim's loss location).

Workaround: For this search, consider the **Reset** button an **Initialize** button.

Internet Explorer 6 has a memory leak (CC-36552)

Issue: If you are running Internet Explorer 6 (or earlier) and you access the ClaimCenter **Desktop Activities** page and repeatedly click **Activities**, your Internet Explorer memory usage increases dramatically.

Workaround: Upgrade to either Internet Explorer 7 or 8.

Evaluations page has errors after adding a new LOB and upgrading from 3.1 (CC-40249)

Issue: If you have added a new LOB and upgrade to this release, the **Plan of Action → New Evaluation** page will display an incorrect title *New Evaluation* instead of *“LOBName Evaluation*, and the page will not have a *LOBName Cost* field for you to fill in. Editing this page brings up this field, but introduces other problems in displayed cost values and names.

Workaround: If you have introduced new LOBs in 3.1.x, you must edit your PCF files to correct this problem. They must use `ClaimEvaluationDetail` page instead of `ClaimEvaluationDetails<LOB>` page. Also, do not customize the `ClaimEvaluationDetail` page heavily.

After upgrading from ClaimCenter 3.1, the doctor is no longer shown in medical detail exposures (CC-40595)

Issue: ClaimCenter 3.1 displayed the value `Exposure.Doctor`, but later versions of ClaimCenter display the value `Claim.FirstIntakeDoctor`. During upgrade, the data is not migrated from the old field to the new field, resulting in the upgraded ClaimCenter version showing a blank instead of the real value.

Workaround: Add another PCF field in ClaimCenter 5.0.x on the **Medical Details** page below the current **Doctor** field for `Exposure.Claim.FirstIntakeDoctor`. This displays the upgraded data.

Potential to lose *LocationXXX* data during upgrade (CC-41555)

Issue: Guidewire changed the location of the `LocationCity`, `LocationState`, `LocationStreet`, and `LocationZip` fields between ClaimCenter 3.1.x, 4.0.x, 5.0.0, and 5.0.1:

- In ClaimCenter 3.1.x, these fields live on the exposure.
- In ClaimCenter 4.0.x, these fields live as extensions on `MobilePropertyIncident`.
- In ClaimCenter 5.0.0, Guidewire moved these fields into a separate `LocationAddress` object, in the core data model.
- In ClaimCenter 5.0.1, Guidewire changed `LocationAddress` into an extension on `MobilePropertyIncident` (in `extensions.xml`).

Upgrading from ClaimCenter 3.1.x or 4.0.x. Guidewire added a version check that should prevent data loss when upgrading to 5.0.2 from either 4.0.x or 3.1.x. It will look for the `LocationXXX` fields in incident or exposure (depending on whether you are upgrading from 4.0.x or 3.1.x). If any of those fields exist, then the version check will insist the following:

1. There is a `LocationAddress` field, of type `Address`, somewhere in the incident hierarchy
2. That any incidents/exposures with the `LocationXXX` fields set map to incident types that contain the `LocationAddress` field. That is, it checks that all `LocationXXX` fields that contain a value have a `LocationAddress` field to use in migrating the data.

If either of these checks fail, the version check will abort the upgrade and you must then follow the steps outlined in the workaround to solve the problem.

Upgrading from ClaimCenter 5.0.0. If you are upgrading from ClaimCenter 5.0.0, there is already a `LocationAddress` field. However, in 5.0.0, Guidewire placed the field in the core data model, while in 5.0.1 and later it is in `extensions.xml`. So, if your target version `extensions.xml` does not contain a `LocationAddress`

field, then the LocationAddress field could get dropped entirely during the upgrade process, possibly losing data. This is the case that the upgrade version trigger does not address.

Upgrading from ClaimCenter 5.0.1 or later. This is not an issue.

Workaround: Depending on the release from which you are upgrading, you need to do the following:

- **ClaimCenter 3.1.x base release.** In 3.1.x, the LocationXXX fields were in the base data model, on exposure. As part of the upgrade, you will need to decide which exposure types map to which incident subtype. You must map any exposure types that have the LocationXXX fields set to an incident subtype that has the LocationAddress field. This ensures that the upgrade process has a place to move the data.
- **ClaimCenter 4.0.x base release.** In 4.0.x, the LocationXXX fields were in extensions.xml, in the incident hierarchy. You could choose where in the incident hierarchy to put these fields. In upgrading to any 5.0.x version, replace these fields in extensions.xml with LocationAddress, at the same level in the hierarchy.
- **ClaimCenter 5.0.0 base release.** In 5.0.0, the LocationAddress field was in the base data model, in MobilePropertyIncident. If you are upgrading to 5.0.x, you will need to make sure LocationAddress is in the extensions.xml file. Guidewire recommends that you put it in MobilePropertyIncident. However, if desired, it is possible—because it now lives in extensions.xml—to move it to a different place in the incident hierarchy.

Creating a new document from a template results in a *ClassNotFoundException* (CC-42867)

Issue: If you implement the IDocumentProduction plugin in Java, then when you create a new document from a template you may encounter a ClassNotFoundException.

Workaround: Copy the file ClaimCenter/modules/cc/config/plugin/ExternalMappings.properties to ClaimCenter/modules/configuration/config/plugin. Then add the following line to the copied file:

```
gw.api.document.DocumentCreationInfo = gw.api.document.DocumentCreationInfo
```

In addition, copy the file ClaimCenter/modules/cc/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf to ClaimCenter/modules/configuration/config/web/pcf/claim/newdocument/NewTemplateDocumentDV.pcf. In the copied file, replace the following line:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    DocumentProduction.createDocumentSynchronously(DocumentCreationInfo.DocumentTemplateDescriptor
      .TemplateId, gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

with the following:

```
<ButtonInput
  action="gw.api.document.DocumentsUtil.renderDocumentContents(DocumentCreationInfo,
    util.document.DocumentProduction.createDocumentSynchronously(DocumentCreationInfo,
      DocumentTemplateDescriptor.TemplateId,
      gw.api.document.DocumentsUtil.getDocumentCreationParameters(DocumentCreationInfo),
      DocumentCreationInfo.Document))"
  id="CreateDocument"
  value="displaykey.Java.Document.Creation.CreateDocument"/>
```

After upgrade, two duplicate incidents are created (CC-44646)

Issue: If you have two exposures with the same incident type details (for example, auto damage on the same car with the same driver), the database upgrader creates two identical incidents.

Workaround: Guidewire is aware of this issue.

Limitation in reporting when a typecode name contains a comma (CC-45305)

Issue: Due to the limitations of the InetSoft reporting software, issues arise when passing a typelist as a parameter to a drill-down report if one of its typecode names contains a comma. (For example, typelist Loss Cause

contains a typecode with the name *Fall, slip, or trip injury*.) If a typecode name does contain a comma, InetSoft thinks it is an array and sends multiple values. Guidewire has implemented a solution for the Loss Cause typelist in ClaimCenter. However, if you are passing any other typelist as a parameter to a drill-down report and it contains a typecode with a comma, you will experience this issue.

Workaround: Rename any commas from the Name attribute of the typecode.

Problematic calls to SOAP API methods (CC-45378)

Issue: Calling ClaimCenter SOAP API methods from within a messaging plugin can be problematic because the root entity is locked during message processing.

Workaround: If you are upgrading to 5.0.x and your 4.0.x or earlier configuration is calling SOAP API functions from within messaging plugins (or message sinks), you need another approach. Contact Guidewire Support for information on alternatives to calling SOAP APIs from within your messaging integration code.

When editing a check, check portions should not exceed total payments (CC-45520)

Issue: In the **Edit Check** wizard, the fixed-amount check portions are not prevented from exceeding the total of the payments on the multi-payee check.

Workaround: You must make the following configuration change in the `EditCheckWizard.pcf` file: add the method `Wizard.validatePayments` to the `onExit` expression for the *NewCheckPayments* wizard step.

Typelist error message: "contains a typecode with an empty code/name" (CC-46085)

Issue: When starting the server, you may see an error that says a typelist contains a typecode with an empty code or name. This error is mistakenly generated and does not indicate any problem.

Workaround: Ignore this error.

Upgrade fails when upgrading CoverageType typelist that already has a code of: BLDG (CC-46510)

Issue: During upgrade, you receive an exception for the following upgrade trigger: `RecodeCoverageTypecodesTriggers$RecodeCoverageTypecode` and your Coverage Type typelist already has a typekey with code =BLDG

Workaround: Guidewire is aware of this issue.

FNOL wizard not saving selected vehicle (CC-47439)

Issue: The second step of the FNOL wizard does not save the selected vehicle when viewing the **Policy Details** screen and 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).`

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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 info, 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.

Exception stack trace in user interface when *DisplayableException* thrown from Policy plugin (CC-47987)

Issue: A GScript `util` class throws a `DisplayableException` to the policy plugin GScript class which is again throwing another `DisplayableException` with the message from the first exception. The exception message is not user friendly.

Workaround: Add the following parameter to your `config.xml` file:

```
<param name="IntegratedStackTraces" value="true"/>
```

ArithmeticException when exchange rate is set to 0 (CC-48438)

Issue: When you enter an exchange rate of 0 in the New Check wizard, you see an `ArithmeticException` error in the user interface.

Workaround: Add a validation expression (attribute `validationExpression` or `requestValidationExpression`) to the `Transaction_ExchangeRate` element in `ExchangeRateInputSet.default.pcf` and `ExchangeRateInputSet.Check.pcf` files.

Iterator buttons not working correctly in `MatterDetailsDV.pcf` (CC-48879)

Issue: On the `MatterDetailsDV` file, the **Add** button does not hide when not in *Edit* mode. Also, the **Add** and **Remove** buttons do not show when in *Edit* mode, on the `StatusLinesLV` file.

Workaround: Set `lockWhileEditing="false"`. Note that it is not locked.

`ShowNewExposureChooseByCoverageMenuForLossTypes` parameter does not allow a blank value (CC-49426)

Issue: If the `ShowNewExposureChooseByCoverageMenuForLossTypes` configuration parameter is not defined in the `config.xml` file, then the server throws an exception trying to look for at least one loss type in this parameter. This is problematic if you do not want the menu to be displayed.

Workaround: Remove the menu from the PCF files.

Unable to subclass gw.api.quickjump.SpecificClaimCommand (CC-49433)

Issue: If you extend the `gw.api.quickjump.SpecificClaimCommand` class to override the `isPermitted()` method, then the Specific Claim QuickJump command is unavailable. It also can cause an error during server startup.

Workaround: You can create a new QuickJump command that subclasses `com.guidewire.pl.web.navigator.commands.DefaultQuickjumpCommand`, and which delegates its operations to an instance of the `gw.api.quickjump.SpecificClaimCommand`.

Note: If you need to create a subclass of `SpecificClaimCommand`, contact Guidewire Support for sample code attached to issue PL-4061.

Integrity checks allow offsetting payments to be associated to a different check than the payment it offsets (CC-49866)

Issue: Integrity checks for staging tables do not ensure that the two payments pointed to by a `TransactionOffsetOnset` row belong to the same check.

Workaround: You must correct the errors in data mapping, and ensure that each Voided or Stopped payment on a Voided or Stopped check also has an offset payment (with negative amount, in *Submitted* status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

Error when removing claim flags through the Team tab (CC-50159)

Issue: The code to remove the flagged status from a claim in the Team view only works with one claim at a time. The following PCFs allow you to select multiple claims in the LV and then click **Remove Flag**, which results in an error. This can happen with these PCFs: `TeamGroupOtherClaims.pcf`, `TeamGroupPendingClaims.pcf`, `TeamUserClaims.pcf`

Workaround: Modify the `flags` attribute of the **Remove Flag** button in each PCF to read:

```
flags="one CanRemoveFlag"
```

Duplicate descriptions for image/jpg mime types in config.xml (CC-50371)

Issue: In the `config.xml` file, there are two entries under MIME types for JPEG: one for `pjpeg` and one for `jpeg`. This causes a problem as the descriptions of both are identical. If you upload a standard JPEG file and choose the wrong JPEG MIME type, ClaimCenter generates an error.

Workaround: Change the description of `pjpeg` so that duplicate entries do not appear in the MIME type drop-down menu in Studio.

License state drop-down menu in the user interface produces unexpected results (CC-50529)

Issue: If you use the `AddressAutocompleteUtil.getStates()` method to populate license states in the `FNOLVehicleIncidentPopup.pcf` file, then you might see results that are not expected, such as countries or retired typecodes. This occurs when you select the license state drop-down menu in the user interface. What is happening is that `AddressAutocompleteUtil.getStates` is passing `true` to the `getTypeKeys` method when it should be `false`.

Workaround: Use the following:

```
function getStates_Ext(country : typekey.Country) : typekey.State[] {
    var statelist = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = "US"; //default to US
    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
```

```

        statelList.add(eachState)
    }
}
return statelList.toArray() as typekey.State[]
}

```

Include this function in your PCF file. Do not call the `AddressAutocompleteUtil` method in the `valueRange`, instead, call this method:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

Incorrect additivity statements in the default logging.properties file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and, therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change the `log4j.additivity` settings in the following sections:

Section	Original Setting	New Setting
All Plugins	<code>log4j.additivity.PluginsLog=false</code>	<code>log4j.additivity.Plugin=false</code>
All API calls	<code>log4j.additivity.APILog=false</code>	<code>log4j.additivity.Api=false</code>
Database	<code>log4j.additivity.DatabaseLog=false</code>	<code>log4j.additivity.Server.Database=false</code>
Messaging	<code>log4j.additivity.MessagingLog=false</code>	<code>log4j.additivity.Messaging=false</code>
ClaimNumGen	<code>log4j.additivity.ClaimNumGenLog=false</code>	<code>log4j.additivity.Plugin. IClaimNumGenAdapter=false</code>
PolicySearch	<code>log4j.additivity.PolicySearchLog=false</code>	<code>log4j.additivity.Plugin. IPolicySearchAdapter=false</code>
ContactSearch	<code>log4j.additivity.ContactSearchLog=false</code>	<code>log4j.additivity.Plugin. IContactSearchAdapter=false</code>

Duplicate check search velocity template is incorrect (CC-50688)

Issue: There is a bug in the Velocity template used to construct the query for finding duplicate checks in the check wizard. It does not handle all the cases where one of the `ServicePdStart/ServicePdEnd` fields is null on one check but not on the other. Usually this does not matter because both these fields are set together so they are both either set or not set, in which case you can use the existing template.

Workaround: Find the following lines in the `check.vm` file in Studio under `configuration` → `Other Resources` → `duplicate-search`. The template can be changed to be more consistent. Instead of:

```

#if ($Check.servicePdStart)
AND cc_check_head.ServicePdEnd > $Util.sql.getSQLStringValue($Check.servicePdStart)
#end
#if ($Check.servicePdEnd)
AND cc_check_head.ServicePdStart < $Util.sql.getSQLStringValue($Check.servicePdEnd)
#end

```

The template can be changed to:

```

#if ($Check.servicePdStart && $Check.ServicePdEnd)
AND cc_check_head.ServicePdEnd >= $Util.sql.getSQLStringValue($Check.servicePdStart)
AND cc_check_head.ServicePdStart <= $Util.sql.getSQLStringValue($Check.servicePdEnd)
#end
#if (!$Check.servicePdStart || !$Check.ServicePdEnd)
AND (cc_check_head.ServicePdStart IS NULL OR cc_check_head.ServicePdEnd IS NULL)
#end

```

The adjusted version compares check service periods if both start and end values are set. Otherwise, it assumes all service periods with either only one bound or neither bounds set match. So checks with complete service periods (both lower and upper bound set) match if the service periods overlap. Checks with incomplete service periods all match (at least as far as their service periods are concerned).

Since this template is configurable, you should not take these changes if you have already customized this logic to your business requirements. Also, you do not need to make these changes if you are satisfied with the current matching behavior.

Claim association can result in database consistency check failure (CC-50724)

Issue: Making a claim association from the user interface can result in making two claims as primary. This results in database consistency check failure.

Workaround: In the user interface, edit associations and make *only one* claim as primary.

Paging controls are disabled while trying to reassign an activity (CC-50899)

Issue: In the user interface, the paging controls become disabled when you try to reassign an activity by searching for a queue.

Workaround: In Studio, set `startInEditMode=true` in the `AssigneePickerPopup`. This is located in the **Advanced Properties** section.

Group cache needs correct sizing (CC-51545)

Issue: Guidewire discovered assignment performance issues during the ClaimCenter 5.0.5 performance testing on a 2.5M claim database due to the fact that the default configuration of the `GroupCache` was not large enough to hold all the groups. This resulted in frequent premature evictions from the cache during assignment and subsequent thrashing of the cache.

Workaround: Increasing the cache size to make it large enough to hold all the groups, 2200 in the test case, improves assignment performance significantly. Guidewire recommends that you add a `GroupCacheSize` configuration parameter to the `config.xml` file and make it large enough to hold all groups.

The `ExceedsAvailableReserves` method on the `CheckSet` entity does not work when editing a check (CC-57426)

Issue: The `ExceedsAvailableReserves` property on the `CheckSet` entity does not work correctly when you edit a check. For this issue to be problematic, you must have explicitly used `CheckSet.ExceedsAvailableReserve` in rules, which is only useful when the configuration parameter `AllowPaymentsExceedReservesLimits = true`. With the default setting of `false`, the Check wizard ensures that a check's payments do not exceed available reserves, including in edit mode.

Workaround: Guidewire is aware of this issue.

Restored claims fail if there is no assignment and the default owner is not in a group (CC-57542)

Issue: There can be problems assigning restored claims if the default owner is not a member of a group.

Workaround: The default owner must belong to at least one group.

RecoverySet can have at most one uncanceled recovery consistency check error (CLM-512)

Issue: Multiple voided recoveries contained within the same transaction create reports of database inconsistencies. This is because the SQL query on the consistency check is incorrect.

Workaround: You can safely ignore the consistency check.

AuthoringDate field is null if ClaimCenter creates a Claim Flag Change note. (CLM-11002)

Issue: If you create a note with a subject of Claim Flag Change, the note's date always appears to be the most recent even if that is not true. This is because there is no authoring date.

Workaround: Use the following code Claim Preupdate rule to correct this error:

```
uses java.util.Date;

for(n in claim.Notes)
  if(n.New){
    n.setFieldValue( "AuthoringDate", new Date() )
  }
}
```

Deleting a user who owned an archive claim and running the archive worker item throws an exception (CLM-12949)

Issue: If a user is deleted after a claim has been archived and restored, then the archive worker item throws a *DBDuplicateKeyException* exception.

Workaround: Set the user who owns an archive claim to *inactive* instead of deleting them.

Some integrity checks do not report LUWID for bad rows (CLM-13028)

Issue: Some of the integrity checks called during staging table loading do not report the LUWID of bad rows,. It is reported as null. This prevents the automated loading program that you may write from excluding bad data from each load batch, making loading more difficult.

Workaround: Perform one of the following:

- Do incremental Integrity Checking after inserting each claim to the staging tables. After inserting the data for each claim, run `ITableImportAPI.integrityCheckStagingTableContents()`. If there is an error, you know that the latest claim/LUWID is bad, and you can insert a `cc_loadexclusion` row for it. Integrity checks are faster than the phase of actual loading the staging data into the production tables, so this should be faster than loading each claim one at a time.
- Optimistic import: Insert all claims in the batch, and run `IntegrityCheckAndLoad`. If there is an error, back off and insert them one at a time. Clear the staging tables and start the batch over, according to the previous bullet. If most of your data is clean, you might consider this approach for efficiency.

Clicking the Edit button for a policy if the Policy entity is configured with InternalOnly fields brings up prompt instead of a confirm dialog box (CLM-13224)

Issue: If you click the Edit button for a policy when the Policy entity is configured with `InternalOnly` fields, the user interface displays a prompt instead of a confirm dialog box.

Workaround: Make the following edits to the `ClaimPolicyGeneral.pcf` file.

Note: Before you can make the following edits to the PCF file in XML, you must first open the file in Studio and make a minor edit to it. When you click Yes to the message asking if you want to save a copy of the file, ClaimCenter Studio saves a copy in `modules/configuration/config/web/pcf/claim/policy/ClaimPolicyGeneral.pcf`. You can open this file in an editor and make the following changes to its XML.

Remove the button with ID `ClaimPolicyGeneral_EditButton3` and replace it with the following.

Note: You must create display keys for the hard coded labels and use them instead.

```
<ToolBarButton
  hideIfEditable="true"
  id="ClaimPolicyGeneral_EditButton5"
  label="displaykey.Button.Edit"
  visible="Claim.Policy.Verified and HasInternalOnlyFields and
  (perm.Policy.makeeditable(Claim) and perm.Policy.edit(Claim))">
  <MenuItem
```



```

        action="gw.api.policy.ClaimPolicyMakeEditableUtil.makePolicyEditable(CurrentLocation,
Claim, true)"
        confirmMessage="displaykey.JSP.ClaimPolicyGeneral.EditEntireVerifiedPolicy"
        hideIfEditable="true"
        id="EditFullPolicy"
        label=""Edit Full Policy""
        visible="perm.Policy.makeeditable(Claim)"/>
    <MenuItem
        action="CurrentLocation.startEditing()"
        hideIfEditable="true"
        id="EditInternalOnly"
        label=""Edit Internal Only""
        visible="perm.Policy.edit(Claim)"/>
    </ToolBarButton>

```

User is allowed to delete a queue that has open activities (CLM-14576)

Issue: If you delete a queue that has activities assigned to it, the activities cannot be reassigned to a user. The Group Admin was allowed to delete a queue that had an open activity. Since the queue is now retired, the activity is now visible only from the **Workplan** screen. The activity can no longer be taken from the queue, since the queue is no longer visible. This means that the claim can never be closed since it has an open activity that cannot be skipped or completed.

Workaround: To prevent accidental deletion of a queue to which one or more open activities are assigned, use a **ToolBarFlag** to control availability of the **Remove** button in GroupQueues detail view. The **Remove** button can be available *if and only if* the user has selected one or more queues that have no open activities.

An exception occurs if you add a second attachment to an email (CLM-14601)

Issue: You cannot add a second attachment to an email. You can add email attachments only if the **IDocumentMetadataSource** plugin is enabled.

Workaround: If you need to add another attachment, then you must create another email and add the second attachment to it.

Acrobat sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in **SampleAcrobat.pdf.descriptor** and **SampleAcrobat.pdf** files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Transaction method runWithNewBundle Does Not Follow Links (PL-1716)

Issue: The following GScript API runs a block of code with a new bundle:

```
gw.transaction.Transaction.runWithNewBundle(block)
```

This method has some limitations. Even if you call `newbundle.add(yourentity)`, the `add` method does not follow the graph links to other entities to which it links. You must manually add each subobject to the new bundle by using the call `bundle.add(linkedEntity)` for each entity. You must add all foreign key references to other entities as defined in the built-in data model files. Additionally, if you have made data model extensions that include foreign key references, you must add them as well.

Workaround: You can commit all entities in the current bundle instead of using this API. This may be an issue when designing implementing a custom web service. Committing all entities in the current bundle includes all entities serialized into the bundle as part of the web service request because they were parameters to a web service method. There might be cases where there are entities that are web service API parameters that you do not want commit to the database.

One workaround is to redesign the web service API to avoid taking entities you do not want to commit by taking GScript class instances instead of Guidewire entities. Your custom GScript classes can provide only the fields that you want to pass in your integration code, which may be a much smaller subset of fields in some cases.

Because they are GScript classes and not entities, committing the entire bundle with `gw.transaction.Transaction.getCurrent().commit()` will skip that data that you choose not to commit to the database.

In Studio, clicking Enter in a display key imbeds '\n' in the display key, which breaks Claim Print (PL-5560)

Issue: In Studio, pressing the Enter key while editing a display key embeds a '\n' in the display key. Hard returns in display keys break the Claim Print functionality.

Workaround: Do not add Enter keys to your display keys if you want to use print functionality.

Invalid activity pattern referenced in a workflow prevents ClaimCenter from starting (PL-5987)

Issue: If a workflow configuration file has an `ActivityStep` that references an activity pattern that does not yet exist in the system or database, then the application fails to start. For example, the activity pattern could be created after deployment in a real production release.

Workaround: Do the following:

1. Remove all the references to the workflow activity pattern
2. Deploy
3. Create the activity pattern
4. Re-deploy the application with the references to the workflow activity pattern in place.

InetSoft Report Scheduling is disabled in the base configuration (PL-8687)

Issue: InetSoft report scheduling is disabled in the base configuration. If the scheduler is enabled for autostart by default, it interferes with restarting of the sree server if the scheduler is not shutdown before restarting the sree server.

Workaround: To re-enable the scheduler, you must edit your `sree.properties`. Change the following properties:

- `schedule.auto.down=true`
- `schedule.auto.start=true`

After the scheduler is enabled, ensure that the property `scheduler.classpath` is set correctly. Refer to the InetSoft documentation regarding the Scheduler for more information. As of sree v.10.0, you can find this information under Section 9: Scheduler of the Administration Reference.

Note: The property `repository.audit.enabled` has been set to `false`, overriding the default InetSoft sree value:

`repository.audit.enabled=false`

This setting disables audit records, which can cause issues with reporting if auditing is not setup properly.

Clicking the back button on a browser should show a warning popup if `MaxBrowserHistoryItems = 0` (PL-9321)

Issue: Because Guidewire applications do not formally support or control browser Back button behavior, the Back button should not be used for navigation in Guidewire applications. However, Guidewire applications do not automatically disable a browser's Back button.

Workaround: If you want to add additional safeguards against accidental use of the Back button, do so by using standard ways to disable these buttons, including:

- Launching the application in a browser frame that has the Back button removed.

- Writing some additional Javascript to pop up a dialog with text saying not to use the **Back** button and giving the option to cancel the **Back** command.

GScript bug in anonymous inner classes implementing an inner interface (PL-9363)

Issue: GScript has a known issue in this release with implementing a Java inner interface in GScript.

Workaround: If the interface has exactly one method, you can take advantage of GScript blocks to implement the interface as a block. The parameters of the block are the same as the parameters to the single method, and the return type is the same as the return type of that method.

For example, the `PluginCallbackHandler` provided by the `MessageReply` interface requires you to implement `PluginCallbackHandler.Block`, an inner interface. This interface has one method, `void run()`, so you can implement it with a block that takes no arguments and has no return value.

However, the following code throws an exception:

```
public function messageReceived(final messageId : int) {
  var myBlock : PluginCallbackHandler.Block = new PluginCallbackHandler.Block() {
    public function run() : void {
      try {
        var message = _messageFinder.findById(messageId)
        message.reportAck()
      } catch(e) {
        _logger.logError("Found exception while acking " + e.printStackTrace())
      }
    }
  }
  _callbackHandler.execute(myBlock);
}
```

You can replace the previous code with the following code:

```
public function messageReceived(messageId : int) {
  _callbackHandler.execute(\ -> {
    try {
      var message = _messageFinder.findById(messageId)
      message.reportAck()
    } catch (e) {
      _logger.logError("Found exception while acking " + e.printStackTrace())
    }
  })
}
```

Email plugin has intermittent transmission failures with incorrect address message (PL-10549)

Issue: The Email plugin intermittently fails to send emails and reports that the message address is invalid, even if the address is actually valid. This intermittent problem has been reported in a small number of cases.

Workaround: Restarting the application server usually resolves the message address problem. If this is a recurring issue, then Guidewire recommends that you implement your own email message transporter. This message transporter must have the ability to read the fields from the email object to retrieve the contents of the email.

Surge in memory usage when connecting multiple instances of Studio to running application (PL-11566)

Issue: A surge in memory usage occurs if you attempt to connect multiple instances of Studio to a running application.

Workaround: Limit the number of Studio instances connected to a server in any shared environments.

Error retrieving replets from report server when you sync from the Report Admin (PL-11913)

Issue: If you click the **Sync** button for the first time in the Report Admin (**Administration** → **Report Admin** → **Reports**) you may see an error when retrieving replets from the report server.

Workaround: You must modify the InetSoft web.xml to set load-on-startup to true for the *ServletRepository* with the following:

```
<servlet>
  <servlet-name>replets</servlet-name>
  <servlet-class>inetsoft.sree.web.ServletRepository</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Deleting a report that is used by a permission set creates an exception in the user interface (PL-12009)

Issue: If you delete a report that is used by a permission set in ClaimCenter, then you will see a Stack Trace error in the **Permission Sets** tab on the permission set that included the deleted report. (The path is **Administration** tab → **Report Admin** → **Permission Sets**.)

Workaround: Remove the report from the permission set before deleting it from InetSoft and the ClaimCenter Reports tab.

Report is still visible in the Report Admin tree after you sync with the Report server (PL-12010)

Issue: If you set a report's visibility to false in InetSoft, you can still see the report in the ClaimCenter Admin Report Tree. The path is **Administration** tab → **Report Admin**. However, these reports still remain hidden to any report users from both the InetSoft Portal and in ClaimCenter.

Workaround: Guidewire is aware of this issue.

Studio does not display complete typecode list during code completion (PL-12137)

Issue: Studio's code completion function displays only a single typecode for a given **Name** value, even if duplicate **Name** values with different typecodes exist in a typelist.

Workaround: Ensure that the **Name** values are unique within each typelist. Or, manually enter the typecode value, which Studio will recognize.

Studio throws exception when creating a top-level display key (PL-14187)

Issue: Studio throws an exception if you attempt to create a top level display key immediately after deleting another top-level display key.

Workaround: Clear the exception from the **Exceptions** dialog and delete the invalid display key in the editor. Then close and reopen the editor and create the display key.

Error when reverting a newly created entity name (PL-14200)

Issue: Studio generates an error if you create a new entity name (**Resources** → **Entity Names** → **New** → **Entity Name**) and then attempt to revert your change in source control.

Workaround: Use the **Delete** option instead of **Revert** and the error does not occur.

chapter 33

Guidewire ClaimCenter 6.0.0 Release Notes

Release 6.0.0.103

Copyright © 2001-2009 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

- This release of Guidewire ClaimCenter is 6.0.0.103
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 10.1, 20090924

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter

The following issues pertain to installing this release.

Inetsoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-798-ERX-0000BF00100001F-F2AFC9DD34ED

If you want to use Inetsoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Upgrade Issues

This section describes changes to the ClaimCenter base configuration that may cause upgrade issues.

Subrogation cookbook implementation can cause upgrade issues (CC-53120)

Issue: ClaimCenter 4.0 customers that implemented the *Subrogation Cookbook*, article ID # 286 on the Guidewire *Customer Service Center Portal*, may run into difficulties when upgrading unless steps are taken prior to upgrade. The cookbook provides to ClaimCenter 4.0 customers similar subrogation functionality to what was released in ClaimCenter 5.0. If you implemented this cookbook without first modifying the entity names listed in the cookbook, then you will not be able to upgrade to ClaimCenter 5.0 or 6.0 without first resolving a naming conflict.

One way to determine if this issue applies is if any of the following entities are found in the `extensions.xml` file: `SubroAdverseParty`, `StatuteLimitationsLine` or `SubroPaymentSchedule`. If you find those entities but they have a slightly different name, such as `subroadverseparty_ext`, then no conflict should occur.

Workaround: If one or more of those entities are found in the `extensions.xml` file, then you have two options.

- Rename the entities so that no conflict exists. For example, change SubroAdverseParty to subroadverseparty_ext. Migrate the data and update all related .pcf files and Gosu code, *or*
- Migrate the data from the current entities to the default configuration entities.

For detailed information on these options, refer to article ID #643 on the Guidewire Customer Service Center Portal.

Reporting Issues

As of this release, Guidewire certifies the reporting application server (InetSoft) to run on Tomcat 6.0.14 *only*. Guidewire explicitly does not certify the reporting server to run on any of the following for this release:

- WebSphere 7.0
- WebLogic 10.3

Guidewire does anticipate certifying Guidewire Standard Reporting on these platforms in a future release.

Reporting Issues With Localized Content

As of this release, InetSoft reports are not fully localized and some reports may contain localization errors in the following areas:

- If a report has a footer, then some of the content in that footer may not be localized.
- The word *Sum* used in some of the *Dashboard* reports is not localized.
- Number formats that are based on a particular country are also not localized.

Guidewire is aware of these issues and will address them in a future release.

Changes in this Release

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes

Configuration and GScript API Changes

For a list of configuration and GScript API changes between ClaimCenter 5.0.5 and ClaimCenter 6.0.0, [click here](#) (requires the readme_files directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- GScript API
- typelists
- display keys

Base PCF File Changes

All links below require the readme_files directory on your local disk.

ClaimCenter release 5.0.5 to 6.0.0

- 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/platform directory, [click here](#).

Known Issues and Limitations

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

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

IClaimAPI depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: IClaimAPI depends on SynchStateData, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

The ContactCenter contact record overwrites ClaimCenter contact information (CC-32732)

Issue: When ContactCenter and ClaimCenter are integrated, ContactCenter contact information overwrites the ClaimCenter contact record.

Workaround: None.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Limitation in reporting when a typecode name contains a comma (CC-45305)

Issue: Due to the limitations of the InetSoft reporting software, issues occur when passing a typelist as a parameter to a drill-down report if one of its typecode names contains a comma. (For example, typelist *Loss Cause* contains a typecode with the name *Fall, slip, or trip injury*.) If a typecode name does contain a comma, InetSoft treats it as an array and sends multiple values. Guidewire has implemented a solution for the *Loss Cause* typelist

in ClaimCenter. However, if you are passing any other typelist as a parameter to a drill-down report and it contains a typelist with a comma, you will experience this issue.

Workaround: Guidewire is aware of this issue and will address it in a future release. If you encounter this issue with any other report parameters, contact Guidewire Support for assistance.

FNOL wizard not saving selected vehicle (CC-47439)

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)"`.

Integrity checks allow offsetting payments to be associated to a different check than the payment it offsets (CC-49866)

Issue: Integrity checks for staging tables do not ensure that the two payments pointed to by a `TransactionOffsetOnset` row belong to the same check.

Workaround: You must correct the errors in data mapping, and ensure that each `Voided` or `Stopped` payment on a `Voided` or `Stopped` check also has an offset payment (with negative amount, in *Submitted* status) on the same check. You can also leave out the offset payment, its `TransactionLineItems`, and the `TransactionOffsetOnset` record, and they will be created automatically during loading.

Exception stack trace in user interface when `DisplayableException` thrown from Policy plugin (CC-47987)

Issue: A Gosu util class throws a `DisplayableException` to the policy plugin Gosu class which is again throwing another `DisplayableException` with the message from the first exception. The exception message is not user friendly.

Workaround: Add the following parameter to your `config.xml` file:

```
<param name="IntegratedStackTraces" value="true"/>
```

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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.

Duplicate descriptions for image/jpg mime types in the configuration file (CC-50371)

Issue: In the `config.xml` file, there are two entries under MIME types for jpeg: one for pjpeg and one for normal jpeg. This causes a problem as the descriptions of both are identical. If you upload a normal 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.

License state drop down menu in the user interface produces unexpected results (CC-50529)

Issue: If you use the `AddressAutocompleteUtil.getStates()` method to populate license states in the `FNOLVehicleIncidentPopup.pcf` file, then you might see unexpected results, such as countries or retired type-codes. This occurs when you select the license state drop down menu in the user interface. What is happening is that `AddressAutocompleteUtil.getStates` is passing `true` to the `getTypeKeys` method when it should be `false`.

Workaround: Use the following:

```
function getStates_Ext(country : typekey.Country) : typekey.State[] {
    var stateList = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = "US"; //default to US
    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
            stateList.add(eachState)
        }
    }
    return stateList.toArray() as typekey.State[]
}
```

Include this function in your `.pcf` file. Instead of calling the `AddressAutocompleteUtil` method in the `valueRange`, call the following method instead:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

Incorrect additivity statements in the default logging.properties file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false
```

to

```
log4j.additivity.Plugin=false
```


The statements for API, Database, Messaging, Messaging.ISO, Plugin.ClaimNumGenLog, Plugin.IPolicySearchAdapter, and Plugin.IContactSearchAdapter are also incorrect and need to be changed.

Tax IDs need to be encrypted before they are written to the database for a claim snapshot (CC-54859)

Issue: ClaimCenter has encrypted tax identification numbers in the default configuration but those fields are not encrypted within the claim snapshot. However, when a claim's snapshot is stored in the database, it too must be encrypted. Currently, it is not so snapshots are not encrypted.

Workaround: If you have enabled claim snapshots, you can refrain from using encryption until the ClaimCenter 6.0.1 release. Since it is enabled in the ClaimCenter 6.0.0 base configuration, you must disable encryption in both the application and in the duplicate check search template. In the application, you must remove (or set to `false`) the encryption column parameter from the TaxID property in the Contact entity and from the BankAccount property in the Check entity. The template (`DuplicateCheckSearchTemplate.gst` file) must also be updated so that it does not encrypt the TaxID entity before doing a search.

You also have the option of leaving encryption enabled and having claim snapshots *unencrypted*. The benefit of this solution is that when Guidewire provides a fix, ClaimCenter will automatically upgrade the unencrypted snapshots.

Auto First and Final Claim wizard displays an exception if you change the policy currency through an independent wizard step (CC-56337)

Issue: In the FNOL wizard, after selecting the policy and continuing in Auto First and Final mode, there are menu options on the left side of the screen to edit the details of the policy. If you edit the policy and change the currency, the new currency is not set in the `NewClaimCheck` object, resulting in an exception in the user interface. This occurs when you click **Add Deductible**. Note that editing the policy and changing its currency from its currency in the policy system makes the policy unverified.

Workaround: After editing the policy's currency, go back to Step 1 of the wizard, then click **Next** to return to Step 2. This updates the `NewClaimCheck` appropriately.

Total approved displays negative amount after editing the bulk invoice item's payment type (CC-53934)

Issue: Editing a bulk invoice by changing a bulk invoice item's reserve line and payment to partial in the same update results in an incorrect amount in the bulk invoice's **Total Approved Amount** field.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Japanese layout issue in Advanced Search screen (CC-56173)

Issue: When the user interface is localized to the Japanese language, the source label at the top goes down the left side instead of across.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Some user interface screens in Japanese are incorrectly rendered (CC-54225)

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: Guidewire is aware of this issue and will address it in a future release.

The localized document pattern specified in the activity pattern is not passed to the activity (CC-55468)

Issue: When ClaimCenter is localized to the Japanese language, and you create an activity using the *Gosu Sample Email Sent Record* activity pattern, the system does not select the correct language.

Workaround: Use the document search feature to find the applicable localized document template.

Integrity and consistency checks on multicurrency amounts are not complete (CC-56468)

Issue: The integrity and consistency checks on multicurrency amounts are not yet as strict as the enforcement that exists when new transactions are created in ClaimCenter.

This is specifically for the entities: `TransactionLineItem`, `Deduction`, and `CheckPortion`. Any two amounts on an entity should be equal when they are in the same currency. For example, for an instance of `TransactionLineItem`, `TransactionAmount` must equal `ClaimAmount` if `Transaction.Currency==Transaction.Claim.Currency`. It is similar for `TransactionAmount-ReportingAmount` and `ClaimAmount-ReportingAmount`. For `CheckPortion`, the enforcement is only performed if `FixedTransactionAmount`, `FixedClaimAmount`, and `FixedReportingAmount` are not null.

Workaround: For multicurrency implementations, manually ensure during the population of staging tables that the amounts on the entities `TransactionLineItem`, `Deduction`, and `CheckPortion` are consistent, given their currencies. For single-currency implementations, all amounts should be the same, avoiding the issue.

Unable to delete a user who owns an archived claim (CC-56351)

Issue: If you add a new user, or any other object in the archiving admin graph, then the next time you archive the claim you will be unable to modify or delete that new admin object until the global cache times out its copy of the object. This happens when the object is newly added to the database.

Workaround: You must wait 60 minutes after archiving a claim before restoring it or performing other operations. The length of time to wait is determined by the `GlobalCacheStaleTimeMinutes` configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt delete the user for 60 minutes. Guidewire may not fix this issue, since performing this operation so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

Exception in the Bulk Invoice screen after restoring and reopening an archived claim (CC-56205)

Issue: If you archive a claim, restore it within 60 minutes of archiving it, and then immediately attempt to view a bulk invoice involving that claim, the system may show an exception message. This is because it takes time for the global cache to be updated after the claim is archived. However, this does not cause any data loss or corruption.

Workaround: You must wait 60 minutes after archiving a claim before restoring it. The length of time to wait is determined by the `GlobalCacheStaleTimeMinutes` configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt to view the bulk invoice for 60 minutes. Guidewire may not fix this issue, since restoring archived claims so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

Archiving rules are needed to prevent archiving of claims associated with bulk invoices or checks that have not been escalated (CC-56370)

Issue: If a bulk invoice is still being edited, processed, or awaiting escalation, any claims associated with it should not be archived. If any of a bulk invoice's claims are archived, when certain operations on the bulk

invoice are attempted, the system generates an error stating that you must restore all archived claims associated with the bulk invoice.

Workaround: Restore the archived claims and retry the operation. Guidewire is aware of this issue and will address it in a future release.

During the restoration of a claim from archive, the EmploymentData entity cannot get the claim currency amount when currency is null (CC-56601)

Issue: Restoration of an archived claim fails if the configuration contains a foreign key extension field that points to the EmploymentData entity, and there are rows in the EmploymentData entity referred to only by such a foreign key in extensions. This is because the EmploymentData entity cannot get to the claim to retrieve the claim's currency amount.

Workaround: Guidewire is aware of this issue and will address it in a future release. Otherwise, you can contact Guidewire Support.

Archived claims can fail if they refer to admin data that has not been synced (CC-54981)

Issue: If you call the method `scheduleForArchive`, then the admin data will not be in sync and the call fails. A race condition occurs because ClaimCenter must sync the admin data used by a claim before it can be archived. Calling the method `scheduleForArchive` puts an item in the queue without any guarantees that the necessary admin data is synced.

Workaround: You can consider a scheduled re-sync of admin data every month or so, based on your archiving frequency and use of the `scheduleForArchive` API to indicate specific claims to be archived. You can also throttle the number of concurrent workers during work hours. This is not available natively, but you can write a program (such as a cron job) to call the APIs.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with some versions of Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Some versions of Internet Explorer do not allow you to set this option, and enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928. However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the ClaimCenter window before closing your browser.

Optionally, you can take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer.

WARNING Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

ClaimCenter displays many warnings when regenerating the toolkit for regenerating Javadoc (CC-55564)

Issue: This issue occurs when running the `regen-toolkit` command.

Workaround: Guidewire is aware of this issue and will address it in a future release.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is a port that the ClaimCenter JMX RMI adapter can use, *if enabled*. The ClaimCenter server then reports a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` through Guidewire Studio.

ISO code example is incorrect in the implementation documentation (CC-56785)

Issue: A code example is incorrect in the Guidewire ClaimCenter Integration Guide. You can find it in the *ISO Integration* topic called *ISO Date Search Range and Resubmitting Exposures and Claims*.

Workaround: If you need the latest version of this example, contact Guidewire Customer Support.

Upgrade trigger is missing for adding metric report permissions sets (CC-56625)

Issue: If you upgrade to ClaimCenter 6.0, there is no upgrade trigger for adding the three report permissions sets used in working with Claim Reports.

Workaround: Since there is no upgrade trigger, you must manually add the three report permission sets from the user interface, specifically from the **Administration** tab. You must then associate the claim reports to those report permission sets, and finally assign them to roles so that users can access the reports. See the section on *Working with ClaimCenter Reporting Permissions* in the *ClaimCenter Reporting Guide* for detailed directions.

The permissions are:

- `viewaggclaimmetrics`, which is assigned to the claims supervisor, manager, and superuser roles
- `viewownmetricalerts`, which is assigned to the adjuster, claims supervisor, manager, and superuser roles
- `viewsupmetricalerts`, which is assigned to claims supervisor, manager, and superuser roles

The Guidewire Studio PCF Editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Workaround: None. Guidewire is aware of this issue and will address it in a future release.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because ClaimCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

Zone data with missing values cannot be loaded (PL-9123)

Issue: Zone data is used for associating postal codes with cities and regions. You cannot load zone data that has empty fields.

Workaround: Guidewire is aware of this issue and will address it in a future release. As a temporary workaround, you can enter something into that field, such as the hyphen (-). Note that when this issue is fixed, you will be able to use empty zones. Objects that rely on zone data such as region, holiday, business week and catastrophes will have to be recreated if you reload zone data to remove the hyphen or change anything in the zone definition. Therefore, you can use this workaround for development, but you must assess if you want to continue to use the hyphens for production or make the necessary changes prior to production.

H2 development database creates LONGTEXT fields as VARCHAR (65000), instead of as CLOB as in Oracle and SQL Server (PL-9314)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database. Note that the H2 database is *only* used for testing and should never be used in production.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

ClaimCenter does not display empty report folders (PL-1281)

Issue: ClaimCenter does not display a report folder in the **Administration** → **Report Admin** page if that report folder is empty.

Workaround: Within InetSoft Enterprise Manager, insert a replet (report template) into the empty report folder. You can then mark this replet as not visible (using the InetSoft Enterprise Manager), which causes the replet to not show in ClaimCenter. However, ClaimCenter does display the parent folder.

Decrypting the Tax ID fails (PL-9665)

Issue: If you change or remove the encryption algorithm in such a way that the encrypted columns shrink, then it causes the resulting upgrade to crash on SQL Server.

Workaround: A workaround is not currently available. However, Guidewire is aware of this issue and will address it in a future release.

Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled (PL-9813)

Issue: This issue concerns the Windows Server 2003 component called *IE Enhanced Security Feature* which, when installed can make the ClaimCenter tabs unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server to resolve this issue.

Country specific field validation does not run if the fields are not modified when a country is changed (PL-9828)

Issue: There may be an error if you change a contact field in the user interface that is validated with country specific field validators. Those specific validators do not get revalidated if the country changes and the field is not modified. This issue only occurs if you use country specific field validators.

For example, a carrier operates in two countries, the United States and Canada, and wants to implement different field validation for certain data types such as Tax ID. If an adjuster changes the country on an existing contact record, from the United States to Canada, without modifying the Tax ID to reflect the new format, the system allows the US formatted Tax ID to be saved. If the adjuster *edits* the Tax ID, then the system revalidates the format.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Entity type display name implementation does not work (PL-9943)

Issue: The mechanism for displaying localized content in an entity type display name does not work. For example, Guidewire provides a localized name for an entity (such as Claim or Company) by adding a line of the form `entity.TypeName = XX` in the `display.properties` file. So `entity.Company = Organisation` would display **Organisation** in the localized content for Great Britain. This means that non-localized type names may display in the user interface. Instead of **Organisation** displaying in the ClaimCenter user interface, **Company** would be displayed. Note that this used to work in previous versions of ClaimCenter.

Workaround: Guidewire is aware of this issue, and will address it in a future release. However, if you need an immediate but limited workaround, you can make edits in Guidewire Studio. Navigate to **configuration** → **Display Keys** → **Web** → **New Contact** → **Title**.

Change:

```
displaykey.Web.NewContact.Title(contactType.TypeInfo.DisplayName)
```

To:

```
displaykey.Web.NewContact.Title((contactType as gw.entity.IEntityType).SubtypeTypeKey.DisplayName).
```

WARNING This workaround is applicable *only* if the code to be fixed is in Gosu and in `.pcf` files. Specifically, this is intended toward a subtype hierarchy. For example the entities **Contact** and **Person** are all parts of the **Contact** hierarchy and you can use this workaround for them. However, **Claim**, **Exposure**, and **Matter**, for example, will not have a `SubtypeTypeKey` so the workaround would not work.

chapter 34

Guidewire ClaimCenter 6.0.1 Release Notes

Release 6.0.1.17

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Insurance Suite, Guidewire ContactCenter, Guidewire Studio, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries. All other product names are trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

- This release of Guidewire ClaimCenter is 6.0.1.17.
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 10.1, 20090924.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter

The following issues pertain to installing this release.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-798-ERX-0000BF00100001F-F2AFC9DD34ED

If you want to use Inetsoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Issues

This section contains types of issues that may affect your installation.

Upgrade Issues

This section describes changes to the ClaimCenter base configuration that may cause upgrade issues.

Subrogation cookbook implementation can cause upgrade issues (CC-53120)

Issue: ClaimCenter 4.0 customers that implemented the *Subrogation Cookbook*, article ID # 286 on the *Guidewire Customer Service Center Portal*, may run into difficulties when upgrading unless steps are taken prior to upgrade. The cookbook provides to ClaimCenter 4.0 customers similar subrogation functionality to what was released in ClaimCenter 5.0. If you implemented this cookbook without first modifying the entity names listed in the cookbook, then you will not be able to upgrade to ClaimCenter 5.0 or 6.0 without first resolving a naming conflict.

One way to determine if this issue applies is if any of the following entities are found in the `extensions.xml` file: `SubroAdverseParty`, `StatuteLimitationsLine` or `SubroPaymentSchedule`. If you find those entities but they have a slightly different name, such as `subroadverseparty_ext`, then no conflict should occur.

Workaround: If one or more of those entities are found in the `extensions.xml` file, then you have two options.

- Rename the entities so that no conflict exists. For example, change `SubroAdverseParty` to `subroadverseparty_ext`. Migrate the data and update all related `.pcf` files and Gosu code, *or*
- Migrate the data from the current entities to the default configuration entities.

For detailed information on these options, refer to article ID #643 on the Guidewire Customer Service Center Portal.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Rules Changes
- *Improvements and General Issues*

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 6.0.0 to 6.0.1

- 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/pl` directory, [click here](#).

Rules Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 6.0.0 to 6.0.1

- To view a report of the changes in the base rules in the `modules/cc` directory, [click here](#).

Improvements and General Issues

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

ID	Description
ABIN-ContactCenter Integration	
CC-57794	Corrected an issue where adding EFT data to a contact would break the integration with between ClaimCenter and ContactCenter.
Administration	
CC-55769	Fixed the catastrophe claim finder query to use IN list for activity status comparison and BETWEEN for claim loss date comparison.
CC-56830	Corrected an exception that was in the user interface when you added a new business week with a zone that already had a business week defined. While ClaimCenter does not allow multiple business weeks in duplicated zones, the fix now provides a meaningful error message so you can understand your mistake and make your correction.
CC-57946	The batch process that is used to match catastrophes with related claims has changed. Refer to the documentation for details.
AGLM-Aggregate Limits	
CC-57424	Fixed <code>NullPointerException</code> s that could occur in the user interface when you added aggregate limits to a claim in the same policy period as an archived claim.
APPL-General Application Bug	

ID	Description
CC-56413	The following ClaimCenter web service implementation classes are now read-only: <code>IClaimAPI</code> , <code>IContactAutoSyncAPI</code> , <code>IExposureAPI</code> , <code>IClaimFinancialsAPI</code> , <code>IBulkInvoiceAPI</code> , <code>IMaintenanceToolsAPI</code> , <code>IMessagingToolsAPI</code> , <code>PCClaimSearchIntegrationAPI</code> , <code>ITemplateToolsAPI</code> , and <code>IDataExtractionAPI</code> . If you want customizations of their behavior, you must create your own customer web service interface using a different name.
ARCH-Claim Archiving	
CC-54981	<p>Improved the JavaDoc for the web service interfaces <code>IClaimAPI</code> and <code>IMaintenanceAPI</code> methods for scheduling claims for archive. It now indicates that there is a possibility of problems if the archive data is not synchronized.</p> <p>Also, Guidewire added a <code>IClaimAPI.scheduleForArchiveByPublicId</code> method. It is identical to <code>scheduleForArchive</code> except that it allows you to specify claims by their public IDs instead of by their claim numbers.</p> <p>In the web services interface <code>IMaintenanceToolsAPI</code>, the method <code>markArchiveReady</code> (which used to just call <code>scheduleForArchive</code> anyway) is now deprecated. Additionally, there is a new method (<code>scheduleForArchive</code>) to explicitly schedule for archive. It is identical to the <code>IClaimAPI.scheduleForArchive</code> method. It is duplicated on <code>IMaintenanceToolsAPI</code> for convenience. The JavaDoc has been improved for that interface to make clear that <code>markArchiveReady</code> is deprecated and internally simply calls the <code>scheduleForArchive</code> method.</p> <p>Guidewire added a <code>scheduleforarchive</code> argument to the maintenance tools command line tool. The admin command line tool scripts are merely wrappers for methods on the <code>IMaintenanceToolsAPI</code> web service interface. You can now run the following command from the command line:</p> <pre>maintenance_tools.bat -server <server> -password <password> -scheduleforarchive -claims <comma separated claim numbers></pre> <p>or</p> <pre>maintenance_tools.bat -server <server> -password <password> -scheduleforarchive -file <file of claim numbers></pre>
CC-56351	Corrected an issue where you were unable to delete a user who owned an archived claim. Now if you restore that archived claim in the base configuration, the claim is assigned to the supervisor of the team to which the deleted user belonged.
CC-56601	<p>Modified the enhancement property <code>EmploymentData.ClaimCurrency</code> to return the default currency if the entity's claim is not accessible. If a foreign key extension field to the <code>EmploymentData</code> entity is added to another entity, and you are configuring a multicurrency implementation, then you must override the <code>EmploymentData.ClaimCurrency</code> property definition.</p> <p>For further details, refer to the property definition in the <code>GwEmploymentDataEnhancement.gsx</code> file or the Gosu API Reference in Studio for this property.</p>
CC-56660	<p>Corrected an issue so that claims that can be purged can now be identified using a purge date. The fix included:</p> <ul style="list-style-type: none"> • The creation of a new nullable property called <code>PurgeDate</code> on the <code>Claim</code> and <code>ClaimInfo</code> entities • creation of a <code>Claim Closed</code> rule that sets the purge date to seven years after the claim close date. • creation of a <code>Claim Reopen</code> rule that clears the purge date. <p>Both rules are not enabled in the base configuration. However, if you do decide to enable them, you must enable both of them.</p>
CC-56878	Corrected an issue where there were potential localization problems when archiving data that contained denormalized fields. If you add extension fields to the <code>Contact</code> , <code>Address</code> , or <code>Claim</code> entities, then you must also add the same fields to the denormalized entities <code>ContactInfo</code> , <code>LocationInfo</code> , and <code>ClaimInfo</code> (respectively) to ensure that these fields are properly archived.
CC-57425	If you archive a claim, ClaimCenter keeps a <code>ClaimInfo</code> stub in the main database, plus some other information such as the claim loss location (<code>LocationInfo</code>). If you added extension fields to, for example, <code>Address</code> then when ClaimCenter archived the claim, the <code>LocationInfo</code> field on <code>ClaimInfo</code> would not have the <code>Address</code> field. The fix included adding a feature so if the same extension field exists on both the main object and the stub object then ClaimCenter copies the extension field to the stub object as well as the core.
CC-57725	The update Archive batch process name and description have been changed to reflect the new archiving framework.
CC-57786	Corrected an issue where you were able to archive a claim with open activities. The fix included changing the archive workitem writer so that it does not pick up claims with open activities. There is also a new rule which skips claims from archiving if there are open activities. See the documentation for further details.

ID	Description
ASGN-Assignment / Segmentation / Strategy	
CC-56239	Added the new method <code>getAllRoleAssignments()</code> to the <code>Claim</code> entity. This method augments the method <code>getRoleAssignments()</code> which filters out any <code>UserRoleAssignment</code> that has an <code>ExposureID</code> . The new method returns all role assignments.
CFIL - Claim File	
CC-54859	Tax IDs are now encrypted before they are written to the BLOB and stored on a claim snapshot. An upgrade trigger that increments the <code>ClaimCenter Encryption Version</code> parameter in the database so that existing snapshots will be upgraded on the fly when they are accessed. Also there is a work queue to upgrade existing snapshots over time to use the current encryption plugin. Refer to the documentation for details.
CC-57546	Fixed an issue where you could not proceed through a validation warning in the user interface when you updated a property incident (<code>FixedPropertyIncident</code>) from an Auto, Property or Traveler's Loss Details screen.
CC-57742	Corrected an issue where if a claim's policy does not have an insured, then the claim's entry in the recently view claims on the Claim tab would show null for the insured named. The fix included modifying the typelist <code>ClaimRecentView</code> so that if the contact name it tries to display is null, then it passes an empty string to the display key instead. You can modify this typelist to reflect a different solution by navigating to configuration → Typelists → ClaimRecentView in Guidewire Studio.
CC-57793	Corrected an issue where if you clicked the Edit button for a policy when the <code>Policy</code> entity is configured with InternalOnly fields, the user interface would display a prompt instead of a confirmation dialog box. The fix for this included replacing the single Edit button with a drop down menu item button that displays two options for a verified policy on a system that has internal fields configured for the <code>Policy</code> entity, to either edit only the <code>ClaimCenter</code> -internal fields, or to deverify the policy and be able to edit all of its fields. If the system is not configured to have internal only fields on <code>Policy</code> , or the policy is not verified, then the Edit button continues to work as it does now.
CC-57955	Corrected an issue where the internal only fields on policies, whose values are supposed to be preserved across a policy refresh, were having their values cleared during that policy refresh. This is because the policy from the system of record does not contain those fields by definition. This means that now, for example, the cancellation date is reset to what it was on the verified policy, but the values of any internal only fields are left as is.
CMDT - Command Line Tools	
CC-56672	Updated some of the batch processes type usage categories (for example, which processes are schedulable, or can be run from either the user interface or from an API. Now you can run Archive, BulkInvoiceSubmission, and Catastrophe Finder from the user interface. The batch processes BulkPurge, BISubmission, Bulk Claim Validation, and Claim HealthCalcs are now schedulable.
CMTX - Claim Metrics	
CC-56441	Corrected an issue where it was not possible to calculate a new metric on old claims, if the old claims already have other metrics. (This was not an issue for new claims.) The fix included adding a new <i>ClaimException</i> rule called <i>CER04000 Recalculate claim metrics</i> . This should only be run when a new metric or indicator has been added and you want to back port them onto the pre-existing claims. In the base configuration, this rule is disabled.
CC-57723	Corrected an issue in the Claim Health metric screen involving the calculation <i>% Reserve Change from Initial Reserve</i> where the cumulative reserves for a claim are divided by the initial reserve amount. This failed if you added 10x more reserves than what was initially set (which is 1000%, exceeding the precision of three digits). The issue was the <code>PercentageMetric</code> datatype. In the default configuration, the application has a scale/precision of (3,0). The fix was to change the <code>PercentageMetric</code> datatype to have a precision of 8 and scale 0.
ContactCenter	

ID	Description
CTC-48	Removed the following batch processes which are not applicable to ContactCenter: <ul style="list-style-type: none"> • Activity Escalation • Geocode Writer • Group Exception • Statistics • User Exception • Workflow
Core	
PL-8740	Guidewire has added two new methods to the IDatabaseUpgrade plugin: <pre>function preUpgrade(context : IUpgradeContext) function postUpgrade(context : IUpgradeContext)</pre> Use these methods as hooks to perform custom SQL before and after an upgrade. If you have already implemented IDatabaseUpgrade in order to upgrade foreign key entities to edgeForeignKey entities, then you still need to implement these new methods. However, you can make them non-operational.
PL-8746	Guidewire has corrected an issue with regen-datamapping-together and regen-datamapping-split that did not generate mappings for custom entities and extensions.
PL-8761	Corrected an issue with query generation that caused Oracle SQLException: <i>identifier is too long</i> . The exception occurred if the query contained an alias longer than the 30 characters allowed by the Oracle database.
PL-9046	Guidewire has modified the localization.xml file to use an XML namespace, rather than a hard-coded path to a schema. The namespace is http://guidewire.com/localization. The schema is located at modules/pl/xsd/localization.xsd
PL-9163	Guidewire has modified the config.xml file to use an XML namespace, rather than a hard-coded path to a schema. The namespace is http://guidewire.com/config. The schema is located at modules/pl/xsd/config.xsd.
PL-9223	Guidewire now provides the ability to generate the Data Dictionary and Security Dictionary as you build the application .war file. Use the following command: gwcc build-war -Dconfig.war.dictionary=true.
PL-9457	Updated the Joda-Time library from version 1.4 to 1.6 to correct a problem with the IBM JDK with WebSphere that returned incorrect results for class DateMidnight.
PL-9685	This Japanese small/large letter equivalence will now only be in effect if a primary linguistic search strength is configured for the Japanese locale. Two small Japanese letters, were added to eight small letters already configured to be treated as equal to the corresponding full size character.
PL-9689	Corrected an issue that occurred when running maintenance_tools -markclaimsforpurge against a claim with a large number of checks (4000). This caused a SQLException.
PL-9793	Guidewire has added the ability to trigger a data model upgrade without modifying any of the data model entity extension files. A new file in Data Model extensions, called extensions.properties, contains a single (numeric) property: version. Guidewire stores the value of this property after an upgrade. As the server starts, the application forces an upgrade if the version in the database is less than the version in this file. The upgrade fails if the version in the database is greater than the version in this file.
PL-9857	Guidewire now provides four modes for debugging when starting the development server during configuration. You can change the debug port to use by configuring the port.properties file. The modes are: <ul style="list-style-type: none"> • dev-debug-shmem (Debug dev-mode webapp using shmem) • dev-debug-socket (Debug dev-mode webapp using socket) • dev-suspend-shmem (Debug dev-mode webapp using shmem, start suspended) • dev-suspend-socket (Debug dev-mode webapp using socket, start suspended)
PL-9914	Guidewire has corrected a localization issue in which the entity type display name did not use the subtype key's localized name. This led to some user interface labels not being localized correctly.
PL-9987	Guidewire has corrected an issue in which running the regen-pcfmapping tool did not correctly pick up custom PCF pages.
PL-10002	Guidewire has provided the ability to add events to an entity via extension. So, if a base entity does not declare events, a customer can add events to the entity by extending the entity and adding the <events> tag.

ID	Description
PL-10027	Guidewire has modified how validation works. Custom entities now trigger validation rules, as long you name the validation rule set correctly. For example, if you create an entity named <code>MyEntityExt</code> , this entity triggers validation rules in the <code>MyEntityExtValidationRules</code> rule set.
PL-10095	Corrected an issue with the MessageTransport plugin that caused it to not shut down if you shut down the application server.
PL-10138	Corrected an issue involving archiving claims with related contacts.
PL-10141	In calling a Java plugin from Gosu code, the application did not automatically convert the domain entity class to an external entity class. For the application to handle this correctly, you need to copy the external entity jar (<code>gw-entity-XX.jar</code>) from: <code>java-api/lib/</code> to <code>modules/configuration/plugins/shared/lib/</code>
PL-10183	<p>Guidewire has modified the behavior of the encryption plugin (<code>IEncryption</code>).</p> <p>It is now possible to register any number of <code>IEncryption</code> plugins, just as with messaging plugins. Only one encryption plugin is the <i>current encryption plugin</i>. There is a new <code>config.xml</code> configuration parameter called <code>CurrentEncryptionPlugin</code>. It specifies which encryption plugin (among potentially multiple implementations) is the current encryption algorithm for the main database. Set the parameter to the plugin name (not the class name). If this configuration parameter is missing, the application uses the <code>IEncryption</code> plugin with name <code>IEncryption</code>. Any legacy existing <code>IEncryption</code> plugins have the plugin name <code>IEncryption</code>. When you use Studio in the Plugins editor to add an implementation of <code>IEncryption</code>, Studio prompts you for a text value to use as the plugin name for this implementation. Guidewire strongly recommends you set the plugin name for encryption plugins to names that describe the algorithm. For example, <code>encryptDES3</code>.</p> <p>Every <code>IEncryption</code> plugin implementation returns the plugin's unique encryption ID as a String value. This is exposed to Gosu as the property <code>EncryptionIdentifier</code>. This API did not change in this release. However, starting in this release this encryption ID is very important. The application decides whether to upgrade the encrypted data with a new algorithm by comparing the encryption ID of the current encryption plugin to the encryption ID associated with the database last time the server ran. If the encryption ID is different, the upgrader decrypts the data with the old algorithm and re-encrypts it with the new algorithm. For new plugins set the encryption ID to describe the algorithm. Do not confuse the encryption ID with the plugin name. The encryption ID is saved in the database. For claim snapshots, the encryption ID is saved also with a claim snapshot.</p> <p>As in previous releases, if the upgrader detects added or removed encryption for any data model fields, the upgrader encrypts the field in the main database using the current encryption plugin.</p> <p>The upgrade-specific plugin interface <code>OldEncryption</code> is now unsupported and removed from the product. Its functionality is replaced by comparing the encryption ID of data in the main database with the encryption ID associated with the current encryption plugin. This triggers decryption and then re-encryption of the data using the newer encryption plugin. The upgrader checks the encryption ID of the plugin to determine whether to decrypt and re-encrypt the main database data.</p> <p>The upgrader for the main database does not upgrade archived claims or claim snapshots. There is a work queue that converts snapshots in the main database but not the archive databases. For details about how ClaimCenter upgrades encryption information for archived claims and claim snapshots, refer to the <i>Encryption Integration</i> topic in the Integration Guide.</p>
PL-10287	Guidewire has corrected an issue that prevented adding a column with a data type of <code>japaneseimperialdate</code> to an entity.
PL-10402	Guidewire has corrected an issue that if you started the tools from <code>Ant/Program.gs</code> , it produced a <code>NoClassDefFoundError: Files</code> on some Windows machines.
PL-10601	Guidewire has corrected an issue that caused the <code>regen-toolkit</code> command to build due to an issue with how the application handled third-party JAR files.
CWIZ-New Claim Wizard	
CC-57184	The Duplicate Claim Search template contained an incorrect query which was fixed. The fix now correctly searches for a claim with the same insured and a loss date within three days of the current claim.
CC-57228	Clarified the documentation (as seen in the Data Dictionary) for the <code>Claim.FirstAndFinal</code> property to indicate that it is not saved to the database, and its value is only valid during the FNOL wizard.
DBUP-Database Upgrader	
CC-53961	Added a version check for duplicate <code>policyproperty</code> and <code>policyvehicle</code> rows. Now <code>CreateRiskUnitsVersionTriggerGroup</code> fails during the <code>CreateAllIndexesVersionTrigger</code> (upgrade) if there are duplicate rows in either <code>policyvehicle</code> or <code>policyproperty</code> .

ID	Description
CC-57777	Updated the entity graph definitions that are necessary during an upgrade. This fixes issues when upgrading from ClaimCenter version 5.0.6 with an archived database.
DMGT-Document Management Subsystem	
CC-56750	Corrected an issue where possibly a document linking to activities and checks using the <code>LocalDocumentMetadataSource.gs</code> code did not work. The issue was that if you enabled the <code>IDocumentMetadataSource</code> plugin and used the provided default implementation, you might have experienced problems attaching documents on certain screens, namely when creating checks and reserves.
EMAIL-Email Support	
CC-55821	Fixed the Canadian French <code>EmailReceived</code> template by renaming the file with the <code>*.gosu</code> extension, creating a descriptor file within the <code>fr_CA</code> directory, and changing the subject value to <i>Email reçu</i> .
EXPO-Exposures	
CC-56759	Fixed a null pointer exception that occurred in the user interface when you removed the <i>sale amount</i> and other fees for a vehicle salvage.
FNLS-Financials Access Subsystem	
CC-43099	Added a consistency check to ensure for every primary check that is voided, pendingvoid, stopped, or pendingstop, there is a corresponding payment on the check.
CC-48326	Fixed the financials calculations to update the entity <code>Checkrpt</code> correctly for <i>Pending Transfer</i> and <i>Transferred</i> checks.
CC-48696	Corrected an issue where you were unable to convert stop/voided checks due to a problem with how the integrity check was generated. Now you can import of voided/stopped checks if <code>AllowMultiplePayments</code> is set to <code>false</code> .
CC-49874	Created integrity and consistency check to verify that a payment and its offset is on the same check.
CC-50291	Corrected an issue where you were unable to transfer a check to another claim with a different claim currency after applying a foreign exchange adjustment.
CC-50316	<p>The methods for applying foreign exchange adjustments were updated for policy-based multicurrency. Separate adjustments can now be made in the claim and policy currencies.</p> <p>Note: if you are upgrading from 6.0.0 there may be a potential issue.</p> <p>The behavior of some foreign exchange methods that were added in 6.0.0 has changed slightly in 6.0.1. Specifically, for those methods that take two <code>BigDecimal</code>-type arguments, passing null for the second <code>BigDecimal</code> argument now indicates that the reporting amount for the check or payment should not be adjusted (rather than indicating, as it did in 6.0.0, that the reporting amount should be adjusted to the same value as the claim amount, as it did in 6.0.0).</p> <p>The affected methods are:</p> <ul style="list-style-type: none"> <code>IClaimFinancialsAPI.applyForeignExchangeAdjustmentToCheck(String, BigDecimal, BigDecimal)</code> <code>IClaimFinancialsAPI.applyForeignExchangeAdjustmentToPayment(String, BigDecimal, BigDecimal)</code> <code>ClaimFinancialsAPIImpl.applyForeignExchangeAdjustmentToCheck(String, BigDecimal, BigDecimal)</code> <code>ClaimFinancialsAPIImpl.applyForeignExchangeAdjustmentToPayment(String, BigDecimal, BigDecimal)</code> <code>Check.applyForeignExchangeAdjustment(BigDecimal, BigDecimal)</code> <code>Payment.applyForeignExchangeAdjustment(BigDecimal, BigDecimal)</code>
CC-50780	Corrected an issue where if you recoded a foreign exchange adjustment payment, it was eroding the reserve.
CC-51091	Fixed transferring checks with foreign exchange adjustment to correctly calculate onset payment's foreign exchange adjustment amounts.
CC-51553	The links Open Claim Financials and Period Financials on the Dashboard tab in the user interface contain calculations of payments and recoveries, such as <i>Total Recovered</i> and <i>Net Total Incurred</i> . Those calculations should not include any offset payment or recovery transactions. The fix removed the offset transactions from the calculations.
CC-52733	<p>Added enforcement in the methods related to check reissuance that the check can actually be reissued.</p> <p>Also, converted the domain method <code>Check.createCheckForReissue</code> into an enhancement method (in <code>GwCheckEnhancement</code>), and added a line to the enhancement method to clear the <code>Comments</code> field on the <code>Check</code> entity on the new check.</p>

ID	Description
CC-53934	Fixed an issue where if you edited a bulk invoice by changing a bulk invoice item's reserve line and payment to partial in the same update results in an incorrect amount in the bulk invoice's Total Approved Amount field.
CC-55120	Added a variant of <code>IBulkInvoiceAPI.createBulkInvoice</code> that allows the caller to specify a custom rate for the new bulk invoice's <code>TransToReportingExchangeRate</code> .
CC-55902	Added database integrity and consistency checks related to the <code>OnsetPublicID</code> field added to <code>TransactionOffsetOnset</code> .
CC-56117	Fixed an issue where the system threw a <code>NullPointerException</code> in the user interface if the deduction's <code>TransactionAmount</code> and <code>ClaimAmount</code> entities are both null.
CC-56219	Corrected an issue where ClaimCenter generated an exception in the user interface if you tried to transfer a check or recovery to an archived claim. The fix included removing the option in the user interface so that you cannot search for an archived claim. The affected files are located in <code>/config/web/pcf/search/claims</code> : <ul style="list-style-type: none"> • <code>ClaimSearch.pcf</code> • <code>ClaimSearchPopup.pcf</code> • <code>ClaimSearchScreen.pcf</code>
CC-56370	A rule, <i>ARC03000 Bulk Invoice Item State Rule</i> , was added to prevent claims from being archived while they are linked to unescalated Bulk Invoice items. By waiting until the items are escalated to archive a claim, you are prevented from having to restore archived claims while working with a Bulk Invoice.
CC-56766	Corrected an issue where an exception was thrown when a reserve was created in the reporting currency but the claim was in a different currency. Also, Guidewire renamed several methods and virtual properties on the <code>TransactionLineItem</code> entity and deprecated their old names: <ul style="list-style-type: none"> • <code>TransactionCurrencyAmount</code> is now <code>TransactionAmount</code> • <code>ClaimCurrencyAmount</code> is now <code>ClaimAmount</code> • <code>ReportingCurrencyAmount</code> is now <code>ReportingAmount</code> • <code>setTransactionCurrencyAmount(CurrencyAmount)</code> is now <code>setTransactionAmountAndUpdate(CurrencyAmount)</code> • <code>updateAmountsFromClaimAmount(CurrencyAmount)</code> is now <code>setClaimAmountAndUpdate(CurrencyAmount)</code>
CC-56872	Corrected an issue where the <code>AbilityToPayResult</code> class was returning messages as pure text strings. The fix now returns messages that may have linkable title pointing to the places in the user interface that failed to validate.
CC-57728	Fixed an issue where you could not add and then immediately remove a document attachment to a check set before clicking Finish in the New Check wizard. This issue also occurred while creating an activity on a claim when you linked a document and then removed it from the list view and then saved the activity.
CC-57734	Fixed a problem with the staging table loader callbacks where the <code>TransToClaimExchangeRate</code> and <code>ClaimToReportingExchangeRate</code> properties of the <code>Transaction</code> entity were not set up correctly for offset transactions that are automatically created for a voided or stopped payment loaded from staging tables.
CC-57815	Corrected an issue where the claim amount, rather than reporting amount, was stored in the <code>FixedReportingAmount</code> property of the entity <code>CheckPortion</code> . This was for <code>CheckPortions</code> created during cancel and reissue of a multipayee check.
CC-57817	Corrected a problem with an integrity check that was previously disallowing all sharing of <code>ExchangeRate</code> entities between transactions in data to be loaded from the staging table. The integrity check now only allows sharing of <code>ExchangeRates</code> among payments on the same check.
CC-58025	Corrected an issue in the Payment Transfer screen where the Net Total Incurred column could display incorrect values or throw an exception.
CC-58269	Changed the <code>Validator.Money</code> display key to be more specific.
FNOL-FNOL Functionality	
CC-47541	Corrected an issue in the auto FNOL where if you did not select a vehicle on step 2, but did select Vehicle Details in step 3, there was an error in the user interface. Now if you select a vehicle's details without first selecting a vehicle, ClaimCenter displays the following message: This vehicle is not listed on the policy.
CC-52690	Corrected an issue where the Incident box on the Loss Details screen in the First Notice of Loss wizard kept growing in size as the text inside kept increasing without wrapping.

ID	Description
CC-56414	Corrected an issue where an <code>IllegalStateException</code> occurred if you changed from using the Quick Claim wizard to the main wizard and then back to the Quick Claim wizard. This was seen in commercial properties lines of business.
CC-56635	Corrected an issue where you saw an empty address in the Loss Location drop down menu when you used the First Notice of Loss wizard out of order.
Gosu	
PL-9275	Studio now generates deprecation reason messages for <code>MetaType</code> properties exposed from <code>TypeLiteral</code> expressions.
PL-9692	Corrected an issue that caused <code>AuthenticationSourceCreatorPlugin</code> implemented in Gosu to fail authentication if used with WebSphere 7.
I18N-Internationalization	
CC-56770	<p>Corrected an error so the following formats for home and work phone are now allowed when adding a contact's number:</p> <ul style="list-style-type: none"> • (123) 1234 5678 or (123) 12345678 • (1234) 123 4567 or (1234) 1234567 • (12345) 12345 • (123456) 12345 <p>Note that a work phone can have an extension such as x1234.</p>
CC-56802	Corrected some labels and typelists on the Contact entity (<code>Contact.txt</code>) where it was displaying incorrect data for localization.
CC-56997	Removed duplicate Country code and added typecode delocalization elements to Country code.
CC-57007	Corrected an issue where if you added a vehicle to an auto claim, it removed all injury incidents that were previously on the claim.
CC-57575	Fixed an issue where the document template descriptors did not accurately use the <code>.xml</code> file format for date. All dates should be formatted in <code>yyyy-MM-dd</code> . Correcting this now means that the document template search returns the correct results.
CC-57967	Changed the <code>NewGroupDetailDV</code> configuration to display the <code>GroupUsersLV</code> label above the LV. If your locale does not evidence the layout problem addressed by this, then this label change can be undone. Also as part of this change, the Member and Manager fields in the <code>GroupUsersLV</code> are now <code>CheckboxCell</code> widgets. They now take up less horizontal space in edit mode compared to the Yes - No radio buttons that currently show for a basic cell widget.
CC-57988	<p>Corrected errors involving adding a new catastrophe. Specifically, this issue is the result of the default <code>CatastropheZonesInputSet.pcf</code> file not providing a valid country (other than the US and Canadian defaults) so that the system can accurately show which zone types are available. Guidewire recommends that you should create a new modal <code>CatastropheZonesInputSet</code> for your country.</p> <p>The fix also included:</p> <ul style="list-style-type: none"> • A change to the default <code>CatastropheZonesInputSet</code> to show an empty list of zone inputs if a country is not defined. Note that actual zone types will only be available once the <code>zone-config.xml</code> file is configured and zones are imported. • Improved handling for populating the zone types list in the Catastrophe Administration user interface. • Updating the default <code>CatastropheZonesInputSet.pcf</code> file to show an empty list of zone inputs if no country has been defined.
Integration	
PL-9353	Part of the Guidewire/InetSoft reporting integration involves a security plugin. Guidewire has modified the plugin implementation to now cache up to one user's information to reduce the number of soap calls made between a Guidewire application and the integrated InetSoft server. The cache has a default timeout of 30 seconds. You can modify this value by altering the property <code>gw.usercache.timeoutinseconds</code> in <code>sree.properties</code> . The value is in terms of seconds.
PL-9569	Guidewire has corrected an issue with reporting JAR file (<code>gw-sree.jar</code>). Previous builds did not automatically contain the Security Provider classes.
PL-9659	Guidewire has modified the base configuration <code>address-config.xml</code> entry for Canada so that autofill works correctly for both city and postal codes.

ID	Description
PL-9714	Guidewire has deprecated the following method: <code>gw.api.webservice.zone.ZoneImportHelper.importDefaultDataToProduction</code> which deprecated the support for the <code>dataFile</code> attribute in the <code>zone-config.xml</code> file.
PL-9721	Guidewire corrected an issue in which disabling the ActiveX document control caused an attempt to create a document from a template to fail. You can now perform this task even if you disable this ActiveX control. This is similar to CC-57112.
PL-9848	Guidewire has fixed the PCF pages for holiday, business week, and catastrophe configuration so that all the zones with granularity configured will show up in the selection.
PL-9900	Guidewire has corrected an issue with incorrect formatting and parsing of the values for XSD <code>gMonth</code> , <code>gDay</code> and <code>gMonthDay</code> .
PL-10599	It is possible that you can experience namespace collisions in the names of argument types and return types on published web services. (If this occurs, Studio notifies you with errors starting in ClaimCenter 6.0.0.) To make debugging during upgrade easier, ClaimCenter 6.0.1 has a new configuration parameter called <code>AllowSoapWebServiceReferenceNamespaceCollisions</code> . If set to true, these error messages become warnings. Use this for development and debugging until you have time to rename your classes to fix the namespace collision. This setting is false by default. Guidewire does not support setting this value to true for production servers. (It is unsafe to do so.) Contact Guidewire Customer Support for complete details.
PL-10629	Guidewire has modified the way it supports InetSoft VPM. This work has tightened the Guidewire SOAP security for <code>ISREEAuthenticationAPI</code> and added three more properties to the <code>sree.properties</code> file. <ul style="list-style-type: none"> <code>gw.soapuser.username</code>: The user name used by InetSoft to retrieve user/group/role information used by VPM and authentication <code>gw.soapuser.userpassword</code>: The password associated with the user name. <code>gw.usercache.size</code>: The cache size for user cache, group cache, and role cache.

ISO Integration

CC-57441	Updated the ISO style sheets. The new ones include the updated content for the CMS changes for Medicare Set Aside.
----------	--

LITM-Litigation Management

CC-48148	Fixed an issue where a litigation history event is no longer created when the Claim's litigation status is set to <i>Not Litigated</i> .
----------	--

Manageability

PL-8055	Guidewire has added a new download to the (System Tools) Data Distribution Info page that reports row counts and growth rates of tables within a specified span.
PL-8108	Guidewire has implemented database searching and sorting appropriate to non-English locales.
PL-8207	Guidewire has modified the colors used on the charts in the (System Tools) Cache Info page to make the information more clear.
PL-8405	Guidewire has added new Oracle-only parameters to the (System Tools) Database Parameters Info page. They are: <ul style="list-style-type: none"> Linguistic Search Options Linguistic search strength Linguistic search function name
PL-9385	Guidewire has corrected an issue that generated a Null Pointer Exception if you clicked the Download button in the (Server Tools) Workflow Info page. The exception occurred if there were any work items that had a status of <i>Error of Failure</i> .
PL-9697	Guidewire has modified the (Server Tools) Data Distribution page so that you can generate data on the minimum and maximum array sizes that shows the row count changes across distributions. This is for all arrays in the data distribution download.
PL-9812	Guidewire has modified the (Server Tools) Database Storage page so you can choose specific tables and mode.
PL-9895	Guidewire has added a (Server Tools) Server DMV Snapshot screen that displays information on the aggregate usage for all queries in the DMV, along with the percentage of resource usage by the top N queries.

ID	Description
PL-9911	Guidewire has modified the (Server Tools) Database Storage page to include the following information: <ul style="list-style-type: none"> • Queries Executed to Build Download • Summary of Queries Executed to Build Download
PL-10000	Guidewire has added two additional columns to the Index Physical Statistics tab of the (Server Tools) Database Storage Information page. The two new columns are fill_factor and is_padded .
PL-10110	Guidewire has modified the (Server Tools) Work Queue Info page to show additional worker summary information, including throughput information.

PADM-Policy Administration / Integration

CC-55297	Added a field on the Policy entity to support handling of an external policy system period for use in cross application reporting.
CC-57867	Corrected an issue where the policy currency should not have been editable without de-verifying a policy. Note that usually editing a policy, de-verifies it. However, there are exceptions. For example, if you added internal only fields to the Policy entity, then you can edit the policy only to change the internal fields and the result is that the policy is still verified. In this case, the currency field is also editable. This is because in the related .pcf files (such as PolicyGeneralPanelSet.XXX.pcf) do not check whether the policy is verified as part of the Editable condition on the Currency field.

PERF-Performance

CC-55903	Modified an index by adding a City field to the Address entity and moved that index to the customer modifiable extension file. This is for the Catastrophe Related Claims batch process performance and you can now change this if necessary.
----------	---

RPTG-Reporting

CC-55892	Ensured that the drill down report visibility was set correctly. You can configure InetSoft report files as either visible or hidden depending on if you want drill down reports to be seen in the user interface. The issue was to fix the enable permissions for all reports so that you could have a report hidden in the user interface, but could still grant access to specific users so that they could run the report. You can modify this in the InetSoft configuration file: repository.xml .
CC-55975	Fixed an issue where the Transaction Detail Group's drill down report used to only list claims if the claim had at least one transaction. Now the report lists a claim even if it does not have a transaction.
CC-56006	Corrected an issue where claim level transactions were not included in Catastrophe Financials reports. The fix included: <ul style="list-style-type: none"> • Adding a new physical view and logical model linking the Claim entity to the Transaction entity (ccpvClaimTransaction, cc1mClaimTransaction) • Repointing the table to the new logical model cc1mClaimTransaction. This captures claim level financials.
CC-56021	The count columns in the five Financials/Monthly reports previously listed the count of transaction sets that had the respective transaction type and cost type, such as claim cost payment. The averages also used this count. Now the count columns reflect the number of claims which contain a transaction set with the respective transaction type and cost type. The Average columns also use the claim count.
CC-56524	Fixed the Exposure By Tier Report for Initial Contact With Claimant to not include any exposures where the DaysInitialContactWithClaimantExposureMetric metric had been skipped. This usually only occurred for non workers' compensation exposures.
CC-56540	Updated the Claim Metric reports so that the text links only appear for the claim number for the Claim Metrics and the metric entry for the Exposure Metrics.
CC-56618	Corrected an issue where the sree.properties file should be configured to have a date format per one locale. By removing the entry for format.date in the sree.properties file, InetSoft will then use the format of yyyy-MM-dd for the date combination in the parameter sheet.
CC-56641	Fixed the Claim Overall Avg report for the Time to First Payment.
CC-56690	Corrected an issue in which Guidewire Standard Reporting displayed the local client currency (meaning the default currency of the operating system) instead of the actual transaction and reporting currency.
CC-56975	You would see a Web Page error when setting values to By Month or By Group drop downs for Claim Metrics reports. Now, all of the claim health metric reports have a default selection for the drop downs.
CC-57599	Fixed an issue in reporting where the Transactions Detail report only displayed Payment transactions, but did not show Recovery, RecoveryReserve, and Reserve transactions.

SECU-Security (Login, Permissions/Roles)

ID	Description
CC-56197	Guidewire added report permission sets to the Reinsurance Manager role. They are: <ul style="list-style-type: none"> reportmanager viewaggclaimmetrics viewownmetricalerts viewsupmetricalerts
CC-58216	Fixed an issue with the claim ACL where ClaimUserRole access was ignoring the claim's AccessProfile.
Studio	
PL-437	Guidewire has changed Studio behavior so that it now flags an empty Rule Condition as an error and displays a red error marker in the right-hand margin of the Condition editor. If you verify the rule, Studio generates a <i>Missing return value</i> error.
PL-8997	Corrected an issue in which Studio failed to correctly create a new Rule Set category in the Resources tree if no rules directory existed in the configuration module.
PL-9214	It is possible to configure Studio to open *.xml files directly in an XML editor that is external to Guidewire Studio. To facilitate this process, Guidewire now provides the following optional XML attribute: <pre>xmlns="http://guidewire.com/datamodel"</pre> This attribute defines the namespace for the XML elements in the file. For information on how to use this feature, see <i>Working with an External XML Editor</i> in the Configuration Guide.
PL-9666	Corrected an issued in which implementing the OldEncryption plugin caused Studio to incorrectly show the plugin under Plugins → gw → plugin → util → OldEncryption as well as Plugins → gw → plugin → util → IEncryption → OldEncryption .
PL-9774	Guidewire has added the ability to change the root entity of a rule. To do so, select a rule set in Studio, right-click, and select Change Root Entity from the drop-down menu.
PL-10020	Guidewire has corrected an issue in which the Studio Rules editor did not properly escape double quote marks in the rule description. This caused rules to not behave as expected and caused Studio to not start upon attempting to restart it.
PL-10185	Guidewire has corrected a Studio issue with searching that threw an error if you were searching within a rule set containing CVS or Subversion files. This issue only occurred if you searched for text or names within the rule resources, not with any other Studio resources.
PL-10293	Guidewire has corrected an issue that caused a Null Pointer Error if you double-clicked a view tab to maximize it.
PL-10329	Corrected a Studio search option that did not work correctly if searching from the lowest node in the rule set hierarchy. You can no longer search on Rule Set View in the global replace dialog.
PL-10333	Guidewire has corrected an issue with the Revert to Base right-click command on PCF folders deleting the base configuration folder.
PL-10357	Corrected an issue that prevented Studio SmartHelp from providing a selection list for Group or User ID if creating assignment rules.
PL-10363	Corrected an issue in which the Studio global replace functionality did not work correctly within rule set categories, generating an error message.
PL-10369	Corrected an issue in which Studio did not properly recognize the VALUE keyword conversionExpression attribute (in a PCF page) causing Studio to display an error message.
TRVL-Travel LOB	
CC-56309	Corrected an issue in the Travel line of business where the Trip Cancellation or Delay Exposure was missing certain fields such as Adjuster, Validation Level, Status, and Create Date. The fix included adding a missing trip cancellation/delay exposure detail view. This detail view contains the reference to ExposureCodingFinancialsInputSet.
CC-56951	Corrected an issue where you could not submit a claim for the Personal Travel policy line incidents when the Country/State information is not in the system. The fix was to remove State in the New Claim wizard.
CC-57885	Fixed an error that showed on the Travel line of business user interface when you added a trip. The fix also included removing a redundant Update button on the trip popup screen.
Web	
PL-8749	Guidewire has added a new configuration parameter: WizardPrevNextButtonsVisible. Setting this parameter to true always renders the Back/Next buttons on a wizard step even if is not available (rendered as grayed-out), to maintain consistent positions of all other buttons in the wizard tool bar.

ID	Description
PL-9307	Guidewire discovered a security vulnerability in a previous ClaimCenter release in which a user who carefully manipulated a server request was able to view more data than that to which the user had legitimate access. This vulnerability was viewed on the Common Vulnerability Scoring System (CVSS) as Medium, with a low likelihood of discovery. Guidewire has modified the application to detect this kind of malformed request and to return permitted data only. Guidewire has incorporated the fix into this release and into all future releases.
PL-9328	Fixed an issue in which a CSV export to Microsoft Excel using an English (UK) locale created an invalid character.
PL-9396	Corrected an issue that caused the menu button label color to be incorrect if user selected a list row.
PL-10130	Guidewire has corrected an issue in which a multi-select input did not correctly display the selected items in read-only mode.

Known Issues and Limitations

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

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.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

Must regenerate database statistics after large batch processing of addresses (CC-29941)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

IClaimAPI depends on SynchStateData, which uses deprecated methods (CC-31611)

Issue: IClaimAPI depends on SynchStateData, which in turn uses deprecated functions like `setMessageSinkID()` and `getMessageSinkID()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, continue to use the deprecated methods.

Link issues between a new contact from the New Check wizard and ContactCenter (CC-32732)

Issue: If you try to link a contact newly created from the New Check wizard to ContactCenter, the contact details can be overridden if the contact already exists in ContactCenter. This occurs if there is a match between the two integrated systems. Rather than creating a new contact, the existing matching contact information in ContactCenter overwrites the contact's information in ClaimCenter.

Workaround: Create the contact first in ClaimCenter (not in the New Check wizard), then if you want, link it to ContactCenter. Finally, use the New Check wizard to create a check for that contact.

SampleAcrobat document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in SampleAcrobat.pdf.descriptor and SampleAcrobat.pdf files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Limitation in reporting when a typecode name contains a comma (CC-45305)

Issue: Due to the limitations of the InetSoft reporting software, issues occur when passing a typelist as a parameter to a drill-down report if one of its typecode names contains a comma. (For example, typelist *Loss Cause* contains a typecode with the name *Fall, slip, or trip injury*.) If a typecode name does contain a comma, InetSoft treats it as an array and sends multiple values. Guidewire has implemented a solution for the *Loss Cause* typelist in ClaimCenter. However, if you are passing any other typelist as a parameter to a drill-down report and it contains a typecode with a comma, you will experience this issue.

Workaround: Guidewire is aware of this issue. If you encounter this issue with any other report parameters, contact Guidewire Support for assistance.

FNOL wizard not saving selected vehicle (CC-47545)

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)".

Additional calls to a policy administration system are made when a duplicate claim check runs (CC-47697)

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 checkForDuplications in the NewClaimWizard.pcf file as seen in the following example:

```
function checkForDuplications() {
    if (Wizard.checkForNewDuplicateClaims()) { NewClaimDuplicationsWorksheet.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.

Duplicate descriptions for image/jpg mime types in the configuration file (CC-50371)

Issue: In the `config.xml` file, there are two entries under MIME types for jpeg: one for pjpgs and one for normal jpegs. This causes a problem as the descriptions of both are identical. If you upload a normal 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.

License state drop-down menu in the user interface produces unexpected results (CC-50529)

Issue: If you use the `AddressAutocompleteUtil.getStates()` method to populate license states in the `FNOLVehicleIncidentPopup.pcf` file, then you might see unexpected results, such as countries or retired type-codes. This occurs when you select the license state drop down menu in the user interface. What is happening is that `AddressAutocompleteUtil.getStates` is passing `true` to the `getTypeKeys` method when it should be `false`.

Workaround: Use the following:

```

function getStates_Ext(country : typekey.Country) : typekey.State[] {
    var stateList = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = "&quot;US&quot;"; //default to US
    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
            stateList.add(eachState)
        }
    }
    return stateList.toArray() as typekey.State[]
}

```

Include this function in your PCF file. Instead of calling the `AddressAutocompleteUtil` method in the `valueRange`, call the following method instead:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

Incorrect additivity statements in the default logging.properties file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```

log4j.additivity.PluginsLog=false
to
log4j.additivity.Plugin=false

```


The statements for API, Database, Messaging, Messaging.ISO, Plugin.ClaimNumGenLog, Plugin.IPolicySearchAdapter, and Plugin.IContactSearchAdapter are also incorrect and need to be changed.

Auto First and Final Claim wizard displays an exception if you change the policy currency through an independent wizard step (CC-56337)

Issue: In the FNOL wizard, after selecting the policy and continuing in Auto First and Final mode, there are menu options on the left side of the screen to edit the details of the policy. If you edit the policy and change the currency, the new currency is not set in the NewClaimCheck object, resulting in an exception in the user interface. This occurs when you click **Add Deductible**. Note that editing the policy and changing its currency from its currency in the policy system makes the policy unverified.

Workaround: After editing the policy's currency, go back to Step 1 of the wizard, then click **Next** to return to Step 2. This updates the NewClaimCheck appropriately.

Japanese layout issue in Advanced Search screen (CC-56173)

Issue: When the user interface is localized to the Japanese language, the source label at the top goes down the left side instead of across.

Workaround: Guidewire is aware of this issue.

ClaimCenter renders some user interface screens in Japanese incorrectly (CC-54225)

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: Guidewire is aware of this issue.

Localized document pattern specified in the activity pattern is not passed to the activity (CC-55468)

Issue: When ClaimCenter is localized to the Japanese language, and you create an activity using the *Gosu Sample Email Sent Record* activity pattern, the system does not select the correct language.

Workaround: Use the document search feature to find the applicable localized document template.

Integrity and consistency checks on multicurrency amounts are not complete (CC-56468)

Issue: The integrity and consistency checks on multicurrency amounts are not yet as strict as the enforcement that exists when new transactions are created in ClaimCenter.

This is specifically for the entities: TransactionLineItem, Deduction, and CheckPortion. Any two amounts on an entity should be equal when they are in the same currency. For example, for an instance of TransactionLineItem, TransactionAmount must equal ClaimAmount if Transaction.Currency==Transaction.Claim.Currency. It is similar for TransactionAmount-ReportingAmount and ClaimAmount-ReportingAmount. For CheckPortion, the enforcement is only performed if FixedTransactionAmount, FixedClaimAmount, and FixedReportingAmount are not null.

Workaround: For multicurrency implementations, manually ensure during the population of staging tables that the amounts on the entities TransactionLineItem, Deduction, and CheckPortion are consistent, given their currencies. For single-currency implementations, all amounts should be the same, avoiding the issue.

Exception in the Bulk Invoice screen after restoring and reopening an archived claim (CC-56205)

Issue: If you archive a claim, restore it within 60 minutes of archiving it, and then immediately attempt to view a bulk invoice involving that claim, the system may show an exception message. This is because it takes time for the global cache to be updated after the claim is archived. However, this does not cause any data loss or corruption.

Workaround: You must wait 60 minutes after archiving a claim before restoring it. The length of time to wait is determined by the `GlobalCacheStaleTimeMinutes` configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt to view the bulk invoice for 60 minutes. Guidewire may not fix this issue, since restoring archived claims so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with some versions of Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Some versions of Internet Explorer do not allow you to set this option, and enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928. However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the ClaimCenter window before closing your browser.

Optionally, you can take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer.

WARNING Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

On AIX without an X11 server, ClaimCenter may report that an X11 resource is missing (CC-5889)

Issue: ClaimCenter does not require an X11 server; however other components of the ClaimCenter environment (such as Java) may inadvertently have this requirement. If so, ClaimCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is a port that the ClaimCenter JMX RMI adapter can use, *if enabled*. The ClaimCenter server then reports a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` through Guidewire Studio.

Guidewire Studio PCF Editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Workaround: None. Guidewire is aware of this issue.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because ClaimCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

ClaimCenter cannot load zone data with missing values (PL-9123)

Issue: Zone data is used for associating postal codes with cities and regions. You cannot load zone data that has empty fields.

Workaround: Guidewire is aware of this issue. As a temporary workaround, you can enter a character into that field, for example, a hyphen (-).

H2 development database creates LONGTEXT fields as VARCHAR (65000), instead of as CLOB as in Oracle and SQL Server (PL-9314)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database. Note that the H2 database is *only* used for testing and should never be used in production.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

ClaimCenter does not display empty report folders (PL-1281)

Issue: ClaimCenter does not display a report folder in the **Administration** → **Report Admin** page if that report folder is empty.

Workaround: Within InetSoft Enterprise Manager, insert a replet (report template) into the empty report folder. You can then mark this replet as not visible (using the InetSoft Enterprise Manager), which causes the replet to not show in ClaimCenter. However, ClaimCenter does display the parent folder.

Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled (PL-9813)

Issue: If you have the Windows Server 2003 component called IE Enhanced Security Feature installed, tabs may be unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server.

Country specific field validation does not run if the fields are not modified when a country is changed (PL-9828)

Issue: There may be an error if you change a contact field in the user interface that is validated with country specific field validators. Those specific validators do not get revalidated if the country changes and the field is not modified. This issue only occurs if you use country specific field validators.

For example, suppose that a carrier operates in two countries, the United States and Canada and wants to implement different field validation for certain data types such as Tax ID. If an adjuster changes the country on an existing contact record from the United States to Canada, without modifying the Tax ID to reflect the new format, the system allows the US formatted Tax ID to be saved. If the adjuster *edits* the Tax ID, then the system revalidates the format.

Workaround: Guidewire is aware of this issue.

In Japanese, the Other Coverage Info Details label displays vertically (CC-50236)

Issue: The Other Coverage Info Details label on the Exposure Details screen displays vertically in Japanese.

Workaround: In Studio, move the Other Coverage Info list view to the page footer.

A null error occurs when a contact subtype is added in certain conditions (CTC-133)

Issue: If you use a different locale other than the default, the type keys without a defined priority may change its natural sort order. The system generates a null error because ContactCenter contained a stale sort order after calculating MatchSetKey.

Workaround: When you imported your other languages through Studio, it should have created an `ABContact.ttx` that represented all the ContactCenter subtype keys, along with their translated names. Add a priority to all of type keys. The priorities should be in the order you would normally prefer them to display if they were in a drop down list (for example, alphabetic). Also, make them unique.

Restoring a claim fails if there is no assignment and the default owner is not in a group (CC-57543)

Issue: A null pointer exception is thrown in the user interface if you removed the default owner from the root group, and you attempted to assign the claim.

Workaround: Put the default owner in a group, preferably the root group (default).

No warning that you will overwrite a Service Provider Review or Question Set (CC-39386)

Issue: If you do not retire the review or question set and import the new review or question set, then there is no warning or mechanism to prevent you from overwriting the existing review or question set.

Workaround: You should create the new review or question set in a new .csv file. Also, ensure that all public IDs of all entities of the new review or question set are unique and different from the existing review or question set. Consider that any entities such as `ReviewType`, `ReviewCategoryQuestionSet`, `QuestionSet`, `Question`, `QuestionChoice`, and `QuestionFilter` could be overwritten.

RecoverySet can have at most one uncanceled recovery consistency check error (CC-57348)

Issue: Multiple voided recoveries contained within the same transaction create reports of database inconsistencies. This is because the SQL query on the consistency check is incorrect.

Workaround: You can safely ignore the consistency check.

User without view note permission is able to view medical note (CC-49923)

Issue: A user without the *view note* permission is able to view medical notes on the **Notes** screen

Workaround: You can add a visibility constraint using permissions to the row iterator displaying notes. This will ensure that medical notes are not displayed to users without the view medical notes permission. Note that the record count will display the complete count with this workaround, even though certain rows are hidden.

Unable to generate a Claim Metrics report through the InetSoft Report Portal (CC-56741)

Issue: The InetSoft onLoad script is designed to be run from within ClaimCenter and it should pass the report the locale. If the report is run outside of ClaimCenter, the locale needs to be set within InetSoft.

Workaround: Run reports through ClaimCenter.

New documents fail to show up in *ActivityDocumentsLV* of *ActivityDetailWorksheet* if ActiveX is disabled (CC-57112)

Issue: If document ActiveX is disabled in the server config file, the newly created documents on an activity failed to show up in *ActivityDocumentsLV* of *ActivityDetailWorksheet.pcf* file.

Workaround: Manually link (using the **Link** button) the newly created document to the *ActivityDetailWorksheet.pcf* file.

Tax ID fields are not masked in claim snapshots that were created in earlier versions of ClaimCenter (CC-58392)

Issue: Tax ID fields in the claim snapshot are not masked in the user interface. This happens because the claim snapshot is using an earlier version of the *ClaimSnapshotContactxxx.pcf* file which does not have the field encryption on the claim snapshot. (xxx represents various versions of the pcf file.)

Workaround: You must manually add a mask to the encrypted fields in earlier versions (such as 5.0.0) on the *ClaimSnapshotContactxxx.pcf* file.

Deleting a report folder from the Report Admin causes an exception (PL-10348)

Issue: If you delete a report folder from the Report Admin, users will not be able to access the Report Admin.

Workaround: Guidewire is aware of this issue.

Server ID for each node must be unique across the entire cluster (PL-11228)

Issue: Running multiple clustered instances of ClaimCenter on a single physical machine can cause a number of problems. In particular, the global cache clean up thread can fail, resulting in excessive garbage collection, poor performance, and possible failure of the instance. This required that you restart the application instance to clear up the problem.

This problem can also occur in a cluster if you explicitly set the server ID of several servers to be the same value.

Workaround: Ensure that each instance in a cluster has a unique server ID. You can set the server ID set through a system property.

Report folders do not display if empty (PL-1281)

Issue: Guidewire ClaimCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in ClaimCenter even if the folder is empty.

Not possible to hot-deploy on all supported application servers (PL-9955)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

Values in MIME-type drop-down menus are not localized (PL-10921)

Issue: ClaimCenter does not localize the values in the **MIME type** drop-down list properly.

Workaround: If you want to localize MIME types, you can do so by modifying the definition in `config.xml`. Currently, ClaimCenter does not support MIME type definitions in multiple languages.

Cancel button does not work properly (PL-10926)

Issue: The **Cancel** button does not work correctly if you start in edit mode by clicking on a button such as **Add** from read-only mode.

Workaround: Use the **Edit** button to move the page into edit mode. In this case, all behavior is as expected.

Finder in annotation causes throws error (PL-10812)

Issue: Studio throws error and fails to do syntax error highlight, code completion, and similar functions, if there is a finder in an annotation. This occurs even if you are connected to the server.

Workaround: Remove the finder from the annotation.

Navigating between pages in wizard loses data (PL-10920)

Issue: In wizard popups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Studio does not properly report verification error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Double quotes in PATH environment variable causes problems (PL-10981)

Issue: ClaimCenter does not run if there are double quotes in the PATH environment variable.

Workaround: Check the PATH variable and remove any double quotes.

Studio Debugger does not handle rule breakpoints properly (PL-10553)

Issue: If you set a breakpoint in a Gosu rule and attempt to step through the code using the Studio debugger, the debugger does not correctly display the code on the breakpoint step.

Workaround: Guidewire is aware of this issue.

Inability to access typecodes that contain a dash (PL-10532)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

Cannot start a Spanish translation pack with the Oracle database (PL-11230)

Issue: If you set the application locale to a locale for which you did not configure database collations (in the `collations.xml`), then the application fails on restart. If this is the case, then you see one of the following error messages (depending on your database):

- `java.sql.SQLException`
- `ORA-12705: Cannot access NLS data files or invalid environment specified`

Workaround: Verify that you have properly configured database collations for a given locale before setting the application to that locale.

Cache summary statistics rely on server ID being unique (PL-11239)

Issue: If you use the same server ID for multiple servers in a cluster, it can create the following problems:

- Different servers in a cluster deleting or changing each others results
- Deadlocks at the database level

Workaround: Guidewire is aware of this issue.

ClaimCenter does not start when using the H2 database (CLM-14548)

Issue: An archive database has been defined in the `config.xml` file for the H2 database. This is incorrect as H2 does not support archive databases and the result is that you cannot start ClaimCenter using the QuickStart method.

Workaround: You must comment out the archive section (in bold) as seen in the following example:

```
<!-- H2 (meant for demo/quickstart use only!) -->
<database name="ClaimCenterDatabase" driver="dbcp" dbtype="h2" autoupgrade="true" checker="false"
  printcommands="false">
  <param name="jdbcURL" value="jdbc:h2:file:/tmp/guidewire/cc"/>
  <param name="jdbcURLtest" value="jdbc:h2:file:/tmp/guidewire/cc/test"/>
  <param name="stmtPool.enabled" value="false"/>
  <param name="maxWait" value="30000"/>
  <!-- <archive name="archive1" driver="dbcp" reference="L">
    <param name="jdbcURL" value="jdbc:h2:file:/tmp/guidewire/cc/archive"/>
    <param name="jdbcURLtest" value="jdbc:h2:file:/tmp/guidewire/cc/archivetest"/>
  </archive>-->
</database>
```

chapter 35

Guidewire ClaimCenter 6.0.2 Release Notes

Release 6.0.2.10

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Insurance Suite, Guidewire ContactCenter, Guidewire Studio, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries. All other product names are trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Display Key Changes for Localized Modules
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

- This release of Guidewire ClaimCenter is 6.0.2.10.
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 10.1, 20090924.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter

The following issues pertain to installing this release.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-798-ERX-0000BF00100001F-F2AFC9DD34ED

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Issues and Major Changes

This section contains issues or changes that may affect your installation.

Clustered Indexes

In this release, Guidewire has implemented clustered indexes on the SQL Server database. The database upgrade automatically recreates non-clustered backing indexes for primary keys as clustered indexes. This change improves the performance of claim archiving and claim purging operations. Before recreating the indexes, the upgrade automatically drops (and later rebuilds) any referencing foreign keys and drops any clustered indexes on tables with a primary key.

Recreating the indexes will cause the database upgrade to take a substantial amount of time – many hours for a large database. Guidewire has tried to contact large SQL Server customer before making this change to ensure it will not cause them problems. But please contact customer support if you have not already been contacted and have a large SQL Server database you plan to upgrade to 6.0.2.

New Report for Claim Currency - Open Claim Financials Dashboard

This version contains a new report: The *Claim Currency - Open Claim Financials Dashboard*. It provides a financial summary in claim-based currency for all currently open claims, organized by Group, LOB, and Loss Type. Refer to the ClaimCenter Report Guide → Dashboard Reports for details.

Archive Functionality

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 implementing archiving to help your company with this analysis.

Linguistic Search Performance

ClaimCenter uses an Oracle Java Stored Function to perform linguistic searches. Such search queries can perform poorly.

See PL-12143 for details.

Upgrade Issues

This section describes changes to the ClaimCenter base configuration that may cause upgrade issues.

Subrogation cookbook implementation can cause upgrade issues (CC-53120)

Issue: ClaimCenter 4.0 customers that implemented the *Subrogation Cookbook*, article ID # 286 on the Guidewire *Customer Service Center Portal*, may run into difficulties when upgrading unless steps are taken prior to upgrade. The cookbook provides to ClaimCenter 4.0 customers similar subrogation functionality to what was released in ClaimCenter 5.0. If you implemented this cookbook without first modifying the entity names listed in the cookbook, then you will not be able to upgrade to ClaimCenter 5.0 or 6.0 without first resolving a naming conflict.

One way to determine if this issue applies is if any of the following entities are found in the `extensions.xml` file: `SubroAdverseParty`, `StatuteLimitationsLine` or `SubroPaymentSchedule`. If you find those entities but they have a slightly different name, such as `subroadverseparty_ext`, then no conflict should occur.

Workaround: If one or more of those entities are found in the `extensions.xml` file, then you have two options.

- Rename the entities so that no conflict exists. For example, change `SubroAdverseParty` to `subroadverseparty_ext`. Migrate the data and update all related `.pcf` files and Gosu code, *or*
- Migrate the data from the current entities to the default configuration entities.

For detailed information on these options, refer to article ID #643 on the Guidewire Customer Service Center Portal.

Display Key Changes

If you have implemented the localized modules, then you must change the display keys in your application. Refer to *Display Key Changes for Localized Modules* for details.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Rules Changes
- Display Key Changes for Localized Modules
- *Improvements and General Issues*

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 6.0.1 to 6.0.2

- 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/pl` directory, [click here](#).

Rules Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 6.0.1 to 6.0.2

- To view a report of the changes in the base rules in the `modules/cc` directory, [click here](#).

Improvements and General Issues

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

ID	Description
Administration	
CLM-13977	Corrected an issue where the Team Aging List View had totals in the list and not in the footer.
CLM-14555	Corrected an issue where the team statistics on the Team tab displayed the incorrect time on the non-batch server.
Archiving	
CLM-12873	Fixed an issue where the claim purge process was not throwing an error if the claim graph was not correct
CLM-14548	Corrected an issue where you could not run ClaimCenter (for example in QuickStart mode) with the default configuration because of an invalid archive definition in the <code>config.xml</code> file.
Claim - FNOL Wizard	
CLM-12869	Corrected an issue where there were potentially duplicate entries for fields and incidents if you jumped around in the New Claim wizard, or you switched between that wizard and the Quick Claim Auto wizard.
CLM-14433	Corrected an issue where the <code>VehicleIncident</code> did not retain association with the <code>VehicleRU</code> if you switched between the Quick Claim Auto wizard and the full wizard.
CLM-14518	In the <code>FNOLVehicleIncidentPopup.pcf</code> file removed the hardcoded screen title string.
CLM-14759	Corrected an issue where there was a potential for deadlock in the special type system ClaimCenter uses for displaying claim snapshot information. This deadlock happened rarely but if it did occur, it could freeze the server, as application server threads would get blocked on the deadlock.
Claim - Summary, Loss Details, Exposures, Incidents, Litigation	
CLM-13109	Corrected an issue where the financials pie chart in the Summary screen displayed the wrong ratio if ClaimCenter was localized for Japanese.
CLM-13120	Corrected an issue where if you created a document from a template and changed the To field, it did not update the document.
CLM-14563	Corrected an issue where the claim number disappeared from the Claim tab when the Print Claim screen was shown.
CLM-14728	Corrected an issue where when querying for notes on a claim, ClaimCenter was repeating part of the query that implemented a security check. The security check is now only included once, instead of three times.
Command-line Tools, Toolkit, Data Dictionary, External Entities, Batch Process	
CLM-14542	Modified the data model for the <code>Claim</code> entity so that the <code>InsuredDenormID</code> column will be populated before the staging table loads. This change results in enforcing the constraint that the claim's insured contact has to be loaded together with the <code>Claim</code> . Imposing the extra constraint provides better performance when importing a large number of claims.

ID	Description
CLM-14733	Added example entries for Platform batch processes (such as the ExchangeRates batch process) in the scheduler-config.xml file. Some batch processes had been previously deleted in this file.
CLM-14753	Modified the data model for the entities Claim and Exposure so that ClaimantDenormID will be populated before the staging table loads. This change results in enforcing the constraint that Claim and the claimant contact property on Exposure has to be loaded together with the claim and exposure. Imposing the extra constraint provides better performance when importing a large number of exposures.
ContactCenter	
CTC-133	Corrected an issue where a null error occurred when a contact subtype was added and a new locale display properties were added to the system.
Contacts, Roles	
CLM-12998	Fixed an issue with the Claim ACL where ClaimUserRole access was ignoring the claim's AccessProfile. This fix also corrects an error in the security-config.xml validation process, so in rare cases where an error was reported but the server still started up, now that error will prevent the server from starting.
Core - Platform	
PL-9041	Modified how the currency amount data type works. You no longer specify secondaryAmountProperty and exchangeRateProperty on columns of type currencyamount. Instead, use the new CurrencyAmountComponent if you want a quantity that is editable in the application interface.
PL-10533	Fixed an issue with the <i>regen-pcfmapping</i> tool that caused it to not generate some display labels.
PL-10601	Corrected an issue that caused the regen-toolkit command to fail due to an issue with how the application handled third-party JAR files.
PL-10749	Provided the ability for extensions to support linguistic search on a base column by providing a supportsLinguisticSearch attribute on the <column-override> element.
PL-10761	Corrected a GosuDoc issue that generated a NoClassDefFoundError exception. Even if a gosudoc.properties file existed in the configuration module, the application did not read it.
PL-10896	Disabled the running of validation rules for importing SystemTables.xml and bootstrap.xml. Guidewire still runs validation rules on files in config/import/gen, which are modifiable.
PL-11001	Corrected an issue in which the gw.api.util.Logger utility did not consistently log the server name to the console and in the log file.
PL-11015	Created a new server mode <i>test</i> , that behaves exactly like a production server, except for the ability to modify the system time through an implementation of ITestingClock.
PL-11053	Dates with trailing characters were previous allowed and parsed with potentially the wrong year. Date strings with trailing characters are now rejected.
PL-11062	Moved gw.plugin.util.RemotableSearchResultSpec and gw.plugin.util.RemotableSortColumn out of util. This change means that they are no longer accessible in Java. Instead, for Java, use com.guidewire.cc.plugin.util.RemotableSearchResultSpec and com.guidewire.cc.plugin.util.RemotableSearchColumn.
PL-11068	In calling a Java plugin from Gosu code, the application did not automatically convert the domain entity class to an external entity class. For the application to handle this correctly, you need to copy the external entity jar (gw-entity-cc.jar) from: java-api/lib/ to: modules/configuration/plugins/shared/lib/
PL-11077	Corrected an issue that caused a Runtime Exception if you did the following sequence of events: <ul style="list-style-type: none"> • Created a user validation rule that references a custom system permission. • Dropped the application database. • Started the application, which triggered a database upgrade.
PL-11078	Guidewire officially deprecates library package gw.api.domain.ServiceInfoSource. In its place, use package library gw.api.system.server.ServerUtil.
PL-11136	Corrected an issue with the configuration upgrader that caused it to throw an exception and quit. This happened specifically if you were running an upgrade within a major release to a maintenance release.
PL-11162	Corrected an issue that caused the application to fail to resize an encrypted column properly if you changed the encryption plugin.
PL-11278	Modified the Data Dictionary so that Yes/No fields reflect the actual column type, which is bit (not Boolean).

ID	Description
PL-11283	Modified the Data Dictionary so that the display name for a foreign key is the <code>columnName</code> attribute for that column.
PL-11284	Changed the Data Dictionary to display the <code>columnName</code> attribute (not the name attribute), if a <code><column></code> element contained both. The dictionary now displays the name attribute if both exist.
PL-11285	Removed the set of links that existed underneath the Subtypes lists (if a data entity had subtypes). The links referenced Fields, Arrays, Subtypes, and Unique Keys. The list of Unique Keys was, indeed, not unique.
PL-11286	Corrected an issue that caused the Data Dictionary to display derived fields and arrays in inappropriate sections.
PL-11312	Corrected an issue that caused the Data Dictionary to reference custom entity database tables incorrectly.
PL-11549	Corrected an issue with the (Server Tools) Guidewire Profiler so that it now correctly displays the proper number of milliseconds. Previously, there were cases in which this value was incorrect.
PL-11634	Corrected an issue in which the application ignored the reject methods on validatable extension entities.
PL-11739	Introduced a way to suppress conversion of containers (arrays, lists, maps) when calling Java code. This only is available if the classes do not use Guidewire entity instances (or containers that contain entity instances). It is also unavailable if the class implements a ClaimCenter plugin interface. To use this feature, underneath the ClaimCenter/modules/configuration/plugins/shared directory, create a new directory called <i>basic</i> . Put your libraries and classes in subdirectories <i>classes</i> and <i>lib</i> , just like other plugins directories.
PL-11819	Corrected an issue with the Studio debugger that caused an exception to occur during the logging of the SQL statements made by Gosu find expressions.
PL-11835	Modified the data model to allow extending a non-extendable entity provided that the only changes are to <code>validator</code> or <code>logicalSize</code> parameters using the <code><column-override></code> element. This usage replace the <code><FieldValidatorOverride></code> and <code><FieldLengthOverride></code> elements, which have been removed from <code>fieldvalidators.xml</code> .

Documentation

- CLM-11349 ClaimCenter 6.0.0 changed some aspects of financials integration but the documentation omitted the details. In ClaimCenter 5.0, some financials status transitions complete only if you marked the message at message creation time using methods on the Message entity, such as `message.submittingCheck(Check)`. Then, message acknowledgment automatically triggered the status change. In ClaimCenter 6.0, the recommended approach was to call new methods on financials objects at message acknowledgment time in your messaging plugins. The transition completes as a result of calling these financials domain methods. Do not use the message-based methods, which continue to work but are now deprecated.

The following methods are new:

- `Check.acknowledgeSubmission()`
- `Check.acknowledgeTransfer()`
- `Payment.acknowledgeRecode()`
- `Recovery.acknowledgeRecode()`
- `Recovery.acknowledgeTransfer()`
- `Recovery.acknowledgeVoid()`
- `Transaction.acknowledgeSubmission()` -- for all transaction subtypes.

If you use the new methods, then remove your references to the deprecated message methods. Do not mix the styles.

The following Message methods are deprecated:

- `submittingCheck(Check)`
- `transferringCheck(Check)`
- `voidingRecovery(Recovery)`
- `recodingPayment(Payment)`
- `submittingBulkInvoice(BulkInvoice)`
- `BulkInvoice.acknowledgeSubmission()`
- `submittingTransaction(Transaction)` - for all transaction subtypes

As a part of this change, there is a minor change to behavior for unexpected status transitions with all the new APIs. If the current status is not the previous status it expects, the new APIs throw an exception. The old APIs would ignore the request. To avoid exceptions, message code such as asynchronous reply plugins can get the status before calling the API. For example, confirm that a check is still in *Requesting* status before calling `check.acknowledgeSubmission()`. Contrast this with the behavior with the deprecated message-based methods if the check status was not *Requesting*, for example if the check updated to *Issued* or *Cleared* already. If you use the message-based methods in such cases during message acknowledgment, ClaimCenter ignores the status change for the Ack (it silently does not transition the check to *Requested*).

ID	Description
CLM-14514	Corrected an issue where it was not possible to calculate a new metric on old claims, if the old claims already have other metrics. (This was not an issue for new claims.) The fix included adding a new ClaimException rule called <i>CER04000 Recalculate claim metrics</i> . This should only ever be run when a new metric or indicator has been added and you want to back port them onto the pre-existing claims. Note that in the base configuration, this rule is disabled.
CLM-14893	There are some minor changes to financials object status APIs. The IClaimFinancials web service interface has four new methods for voiding and stopping checks: stopCheck, voidCheck, stopAndReissueCheck, voidAndReissueCheck. There are four new supported check status transitions: from either status <i>Voided</i> or <i>Stopped</i> to the status <i>Issued</i> or <i>Cleared</i> .
CLM-14920	The documentation changed to clarify some check status transitions, especially unusual status transitions such as pending void to stopped, pending stop to voided, requesting to issued, requesting to cleared, notifying to issued, and notifying to cleared. Guidewire strongly recommends you re-review the check status transition table in the topic: Integration Guide: Financials Integration: Check Integration. Ensure not to attempt any unsupported transitions. Ensure to review the documentation and test your integration code before deploying to production to ensure you handle these transitions appropriately.

Financials - Bulk Invoices

CLM-14447	Fixed an issue where bulk invoices with checks that were in pending transfer or transferred status could not be cleared.
-----------	--

Financials - Checks, Payments, Deductibles

CLM-12864	Fixed an issue with the CheckSet.isExceedsAvailableReserves method returned false when an existing check that was below the available reserve amount was edited to exceed the available reserve amount. The method should now return true whenever the check amount is edited and exceeds the available reserve amount.
CLM-13079	Fixed an issue where you could enter in a check portion fixed amount in a scale greater than the check currency. Now, an error message appears when the system validates payments after the second Payments step to inform you that the problematic additional payee check portion fixed amount scale has to match the check currency scale.
CLM-13150	Corrected the display of the check portion claim currency amount to be in the correct currency scale if the check currency and claim currency were different and the claim currency had a currency scale less than the check currency.
CLM-13412	Previously, you could set TransToClaimExchangeRate on Check and Transaction without recalculating the claim and reporting amounts. The fix included explicitly recalculating the amounts every time a new exchange rate entity is set. The same issue also occurred for ClaimToReportingExchangeRate on Check and Transaction. Now the system recalculates the reporting amounts every time a new exchange rate entity is set.
CLM-14730	Corrected a rare issue where if you did a voidAndReissue operation on a multipayee check, the new CheckPortion entity was created using the amounts from the original check's CheckRpt entity. This led to inconsistencies with the original check's amount and reissued check's amount because the check amount and the CheckRpt amount calculation were different.

Financials - DB Checks - Consistency/Integrity, Staging Tables, Loader Callbacks

CLM-12927	Fixed an issue where the Transaction entity that referred to a new, custom ExchangeRate entity could not be loaded from staging tables if they also referred to the existing, market ExchangeRate entity already in the production database.
-----------	--

Financials - Reserve Lines, TAccounts

CLM-14689	Added an additional method to not evict and reload exposures before committing when using the TransactionWizardHelper.
-----------	--

Financials Miscellaneous

CLM-13185	Corrected an issue where some Financials expressions did not have only committed transactions. The fix included making the AwaitingSubmission expression components available as FinancialsExpressions in Gosu. This allows the creation of custom calculations that include only committed transactions.
-----------	---

Gosu

ID	Description
PL-10778, PL-12585	This change fixes a bug that was introduced in Gosu in ClaimCenter 5.0.6 that caused the arguments of a method call to be evaluated before the object target expression of the method call. In nearly all cases this causes no change in behavior. In extremely rare cases, if there were side effects of the argument expressions that change the object target expression value, the behavior is now consistent with the original semantics.
PL-11044	Added a new server configuration option, called <code>RetainDebugInfo</code> , to facilitate debugging from Studio without a type system refresh. If set to <code>true</code> , <code>GosuClasses</code> does not clear debug information after compilation and the server is always considered to be debuggable.
Infrastructure	
CLM-14511	An informative message now displays if there is concurrent editing or deleting in a ClaimCenter cluster configuration warning users that others had changed data and for you to cancel your changes and retry your change.
Integration - Platform	
CLM-13099	Corrected an ISO integration issue where the Driver License State was mapped to the wrong field: <code>StateProv</code> instead of <code>StateProvCD</code> . ISO does not accept the former mapping.
CLM-13119	Added the following <code>OfficialIDTypes</code> so that there is a one to one mapping between ClaimCenter and PolicyCenter: <ul style="list-style-type: none"> • <code>Code=NCCIintrastate</code> • <code>Name=NCCIintrastate</code> • <code>Description=NCCI Intrastate ID</code>
CLM-13155	Fixed an issue where the ISO <code>CommunicationUseCd</code> from the ACORD XML schema was not mapped correctly.
PL-10503	The update Archive batch process name and description have been changed to reflect the new archiving framework.
PL-10599	It is possible that you can experience namespace collisions in the names of argument types and return types on published web services. (If this occurs, Studio notifies you with errors starting in ClaimCenter 6.0.0.) To make debugging during upgrade easier, ClaimCenter 6.0.1 has a new configuration parameter called <code>AllowSoapWebServiceReferenceNamespaceCollisions</code> . If set to <code>true</code> , these error messages become warnings. Use this for development and debugging until you have time to rename your classes to fix the namespace collision. This setting is false by default. Guidewire does not support setting this value to <code>true</code> for production servers. (It is unsafe to do so.) Contact Guidewire Customer Support for complete details.
PL-11193	Guidewire has added a <code>UseMessageCreatorAsUser</code> property to the <code>emailMessageTransport</code> plugin. By default, the <code>emailMessageTransport</code> plugin will use system user to retrieve a document. Set this value to <code>true</code> to retrieve the document on behalf of the user who generated the email message.
PL-11221	Corrected an issue with the <code>IMessagingToolsAPI</code> API that occurred while attempting to acknowledge messages using web services. This issue occurred if the number of field changes was bigger than the smallest number of field values in each field change.
PL-11548	Corrected an issue with adding more than one document to a <code>ReserveSet</code> if using the <code>IDocumentMetadata</code> plugin. For example, any attempt to add multiple documents to a claim on the New Reserve screen would fail.
PL-11677	Fixed a subtle bug in how messaging destinations reads new messages from the messaging queue and dispatches them to messaging plugins. In a rare edge case, a non-claim-specific message could enter the send queue before the related claim-specific messages because of the separate database queries for these two categories of messages. Starting in this release, in each round of sending, each messaging destination always reads claim-specific messages (safe-ordered messages) from the queue before non-claim-specific messages. (Although the read order changed, the dispatch order to messaging plugins did not change. As before, messaging destination threads send non-claim-specific messages before claim-specific messages.)

ID	Description
PL-12689	<p>Modified the list of file types that ClaimCenter opens automatically. Support has been added for the following file types:</p> <ul style="list-style-type: none"> .docx application/vnd.openxmlformats-officedocument.wordprocessingml.document .xlsx application/vnd.openxmlformats-officedocument.spreadsheetml.sheet .pptx application/vnd.openxmlformats-officedocument.presentationml.presentation .ppt additional extension for application/vnd.ms-powerpoint .pps application/vnd.ms-powerpoint .ps application/postscript .rtx text/richtext .wav audio/wav .wma audio/x-ms-wma .mdi image/vnd.ms-modi .gif image/gif .png image/x-png .mov video/quicktime .mpeg video/mpeg .mpg video/mpeg (not returned) .avi video/x-msvideo <p>Since .avi and .wma files can contain security vulnerabilities, Guidewire strongly recommends that you update all client machines with video-playing software that contains the latest security patches. For specific information, see Microsoft web pages <i>.avi files vulnerability</i> (http://www.microsoft.com/technet/security/Bulletin/MS09-038.mspx) or <i>.wma file vulnerability</i> (http://www.microsoft.com/technet/security/bulletin/ms09-051.mspx).</p>
Manageability - Platform	
PL-10035	Modified the (Server Tools) Data Distribution page and added several new non-Boolean types to track the size of a group of values and the number of groups with that size.
PL-10984	Modified the (Server Tools) Info Pages → Data Distribution screen to ensure that the user selects at least two data distributions to include in the download.
PL-11038	Corrected a caching issue with the Sequence Generator that caused it to work incorrectly.
PL-11127	Modified the (Server Tools) Data Distribution page to disable the Download Zip File button if no background data distribution batch process is running.
PL-11153	Added a new download to the (System Tools) Work Queue page. This download provides a zipped CSV file with the history of a given work queue (timing, completed items count, and throughput).
PL-11244	Added a new (Server Tools) Database Statistics screen that you can use to view previous database statistics based on a given date.
PL-11427	Added a new sqlserverCreateIndexSortInTempDB attribute to the <upgrade> element of the <database> element in the config.xml file. The default is false. If true, the attribute enables the ability to use tempdb for sorting if building indexes in SQL Server. Refer to http://msdn.microsoft.com/en-us/library/ms188281.aspx for details on the requirements to use tempdb for sort results.
Performance	
CLM-14436	Corrected an issue where the simple Claim Search page did not limit the number of search results. This could have lead to performance problems if a search would return a large number of results. The fix includes ClaimCenter displaying a message if more than 300 results are found in the simple Search .
CLM-14497	<p>Added the following configuration parameters to control whether the Team Group Activities query uses certain Oracle optimizer hints:</p> <ul style="list-style-type: none"> DisableIndexFastFullScanForTeamGroupActivities DisableCBQTForTeamGroupActivities DisableHashJoinForTeamGroupActivities DisableSortMergeJoinForTeamGroupActivities <p>These parameters were present in previous versions (5.0.x) of ClaimCenter but had not been merged forward to 6.0.x until this release.</p>

ID	Description
CLM-14580	<p>Fixed an issue where the performance of the Property Incident and Loss Details screens is very slow if there are a large number of properties on the policy. This occurred because the <code>AddressInputSet</code> uses a <code>CCAddressOwner</code> object to decide which address fields to show, which fields are editable and so forth. The input set repeatedly queried the address owner object for the set of editable addresses and, in early versions of 6.0.x, the address owner object was reconstructing the entire set every time it was asked.</p> <p>Now the address owner implementation has been changed to use a more efficient approach to constructing the set of addresses and it also caches the set so that it only needs to construct it once.</p>
CLM-14726	<p>Changed the indexes on the entity <code>Person</code> first and last name fields for performance reasons. As part of this change the indexes have also been moved from a non editable file to the customer extension file <code>Person.etx</code>, so that you can change the indexes if needed. However Guidewire strongly recommends that customers start out using the indexes provided in the <code>Person.etx</code> file and only change them after consulting Guidewire support and/or doing extensive performance testing.</p> <p>If you are upgrading, it is important that you merge these indexes into your <code>Person.etx</code> file. In previous releases, you could not modify these indexes but the indexes also could not be omitted by mistake. Now, you can modify them but they can be omitted due to a bad merge or mistaken edit.</p>
CLM-14758	Introduced a change to an index on the <code>Activity</code> table, for performance reasons. This makes an upgrade necessary, though the change (rebuilding the index) is small and should not take more than a few minutes.
CLM-14782	Corrected an issue where ClaimCenter was not populating the claim search cache correctly. This resulted in two count queries on a Claim Search page instead of just one.
CLM-14854	Modified the default schedule in the <code>scheduler-config.xml</code> file for the Financials batch processes. This fixed a performance issue due to the <code>TAccount Escalation</code> and the Financials Escalation processes running at the same time. Guidewire recommends running <code>TAccount Escalation</code> just after midnight, so financials calculations are up-to-date if accessed by rules during subsequently run batch processes. The scheduling of the Financials Escalation process is entirely configurable, though Guidewire recommends the first run be after all other batch processes have finished in the morning.
PL-10874	Modified how the application handles caching of zone data. Previously, large amounts of data would create a severe performance bottleneck and had the potential to crash the application server. This performance fix greatly mitigates these issues.
PL-11431	Converted all backing indexes for all primary keys to clustered indexes in SQL Server. The change improves performance of archiving and purge operations. During a database upgrade, the upgrader automatically converts the backing indexes for all primary keys to clustered indexes and drops any clustered indexes on tables with a primary key.
PL-12257	Guidewire has modified how the application handles caching of zone data. Previously, large amounts of data would create a severe performance bottleneck and had the potential to crash the application server. This performance fix greatly mitigates these issues.
Printing	
CLM-13046	<p>If you printed a multicurrency claim's financials data, it would not include the claim amount. The fix included adding a new display key to support this fix: <code>Java.Print.Financials.Multicurrency.Format</code></p> <p>This display key is used for formatting the output of the multicurrency value during printing.</p>
Reporting	
CLM-304	The report <i>Group by Coverage</i> was missing in the Dashboards reports. It now shows.
CLM-12858	Fixed a reporting issue where the count did not match the claim count in the main report and in the drill down report.
CLM-12865	Fixed an issue where logging in to view reports through ClaimCenter and the InetSoft portal caused user session errors that resulted in <code>SRSecurityExceptions</code> or the user to be permanently locked out.
CLM-12953	Fixed the sorting order for the <i>The Claim Health Metric by Group</i> reports. This applied to the <i>Claim by Group</i> , <i>Claim by Group and Tier</i> , and <i>Exposure by Tier</i> reports.
CLM-13053	Corrected an issue with ClaimCenter on an Oracle database. if you created a check with a payment address that was the same as a user address, the report for <i>Matching Payment Address</i> contained no data.
CLM-13592	<p>Corrected an issue where claim level transactions were not included in <i>Catastrophe Financials</i> reports. The fix included:</p> <ul style="list-style-type: none"> • Adding a new physical view and logical model linking the <code>Claim</code> entity to the <code>Transaction</code> entity (<code>ccpvClaimTransaction</code>, <code>ccImClaimTransaction</code>) • Reprinting the table to the new logical model <code>ccImClaimTransaction</code>. This captures claim level financials.

ID	Description
CLM-13719	Corrected an issue where the InetSoft report: <i>Catastrophe Overview: Claim count on Catastrophe by State</i> pie chart did not match the number of claims that were listed in the <i>Catastrophe Detail</i> drill down report.
CLM-13735	The count columns in the five <i>Financials Monthly</i> reports previously listed the count of transaction sets that had the respective transaction type and cost type, such as claim cost payment. The averages also used this count. Now the count columns reflect the number of claims which contain a transaction set with the respective transaction type and cost type. The Average columns also use the claim count.
CLM-13844	Added a new report called Open Financials Dashboard - Claim Currency. See the ClaimCenter Report Guide documentation for details.
CLM-14459	Corrected a security issue with InetSoft reports. The fix included using username credentials instead of client information for principal creation.
CLM-14473	Fixed an issue where some ClaimCenter reports were located outside the ClaimCenter directory.
CLM-14769	Corrected an issue in reporting where there was more than one logical model in binding. This caused performance issues.
CLM-14919	Corrected an error where you were unable to generate the <i>Litigated Claim Detail</i> drill down report.
CLM-14921	Corrected an issue where you were unable to drill down into the following reports under the Financials folder: <i>Catastrophe Financials</i> and <i>Catastrophe Severity</i> reports.
CLM-14948	Corrected an issue in reporting where there was more than one logical model in binding. This caused performance issues.
PL-10629	Guidewire has modified the way it supports InetSoft VPM. This work has tightened the Guidewire SOAP security for ISREEAuthenticationAPI and added three more properties to <code>sree.properties</code> . <ul style="list-style-type: none"> <code>gw.soapuser.username</code>: The user name used by InetSoft to retrieve user/group/role information used by VPM and authentication <code>gw.soapuser.userpassword</code>: The password associated with the user name. <code>gw.usercache.size</code>: The cache size for user cache, group cache, and role cache.
PL-10972	The application server now logs an error whenever its connection status to the InetSoft report server changes.
PL-11010	Modified the default behavior of <code>sree.properties</code> configuration parameter <code>dependency.checker.enabled</code> . The default is now <code>false</code> . This disables the InetSoft dependency finder. If you set this parameter to <code>true</code> , it can significantly impact system performance.
PL-10629	Guidewire has modified the way it supports InetSoft VPM. This work has tightened the Guidewire SOAP security for ISREEAuthenticationAPI and added three more properties to <code>sree.properties</code> . <ul style="list-style-type: none"> <code>gw.soapuser.username</code>: The user name used by InetSoft to retrieve user/group/role information used by VPM and authentication <code>gw.soapuser.userpassword</code>: The password associated with the user name. <code>gw.usercache.size</code>: The cache size for user cache, group cache, and role cache.
PL-10972	The application server now logs an error whenever its connection status to the InetSoft report server changes.
PL-11375	Corrected a security issue with InetSoft reports. The fix included using username credentials instead of client information for principal creation.
Search	
CLM-13161	Corrected an issue so now the Claim Search , Activity Search and Payment Search screens do not issue a query with more than 2100 group parameters on SQL Server. If that occurs ClaimCenter displays the following message if the number of groups to search is greater than 2,100: Too many assigned groups. Please narrow your search and choose again.
Studio	
PL-10380	Modified the behavior of the Messaging editor so that the Remove button is unavailable if there is only a single defined messaging environment.
PL-10442	Corrected an issue that prevented Studio from moving a PCF file between folders if its name was not also changed.
PL-10447	Corrected an issue that caused a <code>ClassNotFoundException</code> (on <code>AbstractFireTest</code>) upon trying to start the GUnit Debug utility.
PL-10455	Corrected an issue in which using <code>CTRL+left/right</code> arrow to move the cursor from one word to another treated characters accents, cedillas, umlauts, and other modifiers as word breaks (in the same way it treats spaces). This happened with é, ç, and è, for example.

ID	Description
PL-10530	Added the ability to manage certain CSV (comma separated value) files within Studio. (These files are in the <i>Other Resources</i> folder and include <code>authority-limits.csv</code> , <code>activity-patterns.csv</code> , <code>reportgroups.csv</code> , and similar files.) Studio manages the file (adding an editable copy to the configuration folder, for example). You must still, however, edit these files in an external editor.
PL-10570	Corrected an issue that caused an <code>Illegal ordering found for element InputFooterSection</code> error. This occurred if you added an <code>InputColumn</code> widget to the far right of an existing <code>InputColumn</code> widget in a PCF containing an <code>InputFooterSection</code> and tried to save the change.
PL-10598	Modified the behavior of the PCF <code>TypeKeyInput/RangeInput</code> widget's filter so that it now recognizes <code>VALUES</code> as a valid value.
PL-10605	Modified the behavior of the Studio Entity Names editor so that it now properly displays entities with <code>displayName</code> columns, and makes the symbols used in the Entity Name Gosu editor into string types for <code>displayName=true</code> columns.
PL-10698	Modified the behavior of the Create Display Key dialog so that it now wraps text automatically as needed.
PL-10728	Corrected an issue that caused a <code>ClassNotFoundException</code> error if you attempted to revert a newly created PCF file within source control using the (Studio) File → Source Control → Revert menu.
PL-10734	Modified the Rule Conditions editor to display an error if the condition is not a boolean expression. (For example, an assignment statement such as <code>account.AccountType = "payer"</code> is not a boolean expression and should display an error.)
PL-10806	Modified the behavior of the Go-To (CTRL-N) pop-up dialog so that you can now search on workflow names, entity names, and plugin names.
PL-10835	Corrected an issue in which Studio did not display a warning on the use of improper case in Gosu templates. This occurred, for example, if you defined a variable with an initial upper-case character and then you later used that variable with all lower-case characters.
PL-10919	Modified the (Studio) Messaging editor to disable the Add button if the <code>Env</code> field is empty (meaning no name exists in the <code>Env</code> field).
PL-11023	Corrected an issue with typing text in the (Studio) New Workflow Subtype Metadata dialog.
PL-11024	Corrected an issue that permitted recursive screen references on a PCF page, causing a stack overflow.
PL-11137	Modified the extension file that Studio automatically generates when you select an entity <code>.eti</code> file, right-click, and select the Create extension file option. Studio now creates separate <code><extension></code> and <code></extension></code> tags, making the file easier to modify.
PL-11195	Corrected an issue that caused Studio to truncate the display key text used to generate widget titles in PCF files.
PL-11225	Updated Studio so that the (File) Import and Export submenus are correctly localized, if you install a localization module.
PL-11339	Corrected a timing issue that required you click OK twice to close the Display Key Editor dialog that opened from the PCF editor.
PL-11345	Corrected an issue in which the GUnit tester failed (and generated an error) if you attempted to reinvoke a test from the Result window.

Upgrade: Database or Configuration

CLM-14839	Fixed an issue where due to an error in a database upgrade trigger, the ClaimCenter version from 6.0.0 to 6.0.1 upgrade failed if done on a case-sensitive operating system. This was due to a datatype file not being found.
-----------	---

Web - Platform

PL-10438	Corrected an exception if you revisited an uncommon search page whose search result is not Query Result type.
PL-10476	Added the ability to generate a new row in <code>ListView</code> edit mode by pressing ENTER in the last editable cell.
PL-10706	Corrected an issue that generated a <code>NullPointerException</code> if you called an <code>addFieldError</code> method from a <code>validate</code> method invoked from certain PCF page.
PL-11227	Corrected an issue that the application did not correctly apply the <code>useArchivedStyle</code> property to a <code>TextAreaCell</code> widget. (The cell still shows the standard font color.)

Display Key Changes for Localized Modules

If you have implemented the localized modules, then you must make the following changes to the display keys.

Display Keys for ClaimCenter

The following section affects ClaimCenter customers who have a localized build. You must either add, modify, or delete the following display keys in your customer configuration file, specifically in your locale folder. Make your changes for all the display keys in the applicable file (for example, `display.properties`, or `studio.display.properties` files.)

WARNING If you fail to make these changes to your configuration, then the application will not start.

Add the following display keys in `display.properties`

`Java.Print.Financials.Multicurrency.Format = {0} = {1}`

`Java.Search.TooManyGroups = Too many assigned groups. Please narrow your search and choose again.`

`Web.FNOLInjuryIncidentPopup.Title = Injury Incident`

`Web.FNOLVehicleIncidentPopup.Title = Vehicle Details`

`Java.Database.NoCollationDefined = No collation name found in collations.xml for locale {0}`

`Java.Database.NoSortCollationDefinedForOracle = A specific Oracle collation for database sorting was not configured in collations.xml for the default application locale {0}.`

`Java.Database.NoSortCollationDefinedForOracleDefaultUsed = Collation BINARY_CI will be used instead.`

`Java.Database.UnsupportedDatabaseVersionAllowedInDevMode = Unsupported database version allowed because server is running in development mode`

`Java.Database.UnsupportedJDBCdriverAllowedInDevMode = Unsupported JDBC driver allowed because server is running in development mode`

`Java.Entity.Error.IdentifierWithDisplayName = '{0}':{1} - {2}'`

`Java.Entity.Error.IdentifierWithoutDisplayName = '{0}':{1}'`

`Java.Error.UpdateStatsDVHelper = Update Statistics batch process is already running`

`Java.InternalTools.InfoPages.DataDistribution.ComparisonRequiresTwoOrMore = To download a comparison of data distributions, you must select two or more data distributions.`

`Java.ServerMode.Label.Dev = [DEV mode - {0}]`

`Java.ServerMode.Label.Test = [TEST mode - {0}]`

`Java.ServerMode.Warning.Dev = The server is in "development" mode. This mode should never be used for a production server.`

`Java.ServerMode.Warning.Test = The server is in "test" mode. This mode should never be used for a production server.`

`Web.Admin.WorkflowDetail.Parent = Parent Workflow`

`Web.Admin.WorkflowDetail.SubFlows = Children Workflows`

`Web.Admin.Workflows.Children = Children`

`Web.Admin.Workflows.Parent = Parent`

`Web.DBStatsSubmitDV.CanChoosePointInTime` = A specific point in time can be selected to display statistics that were current at that time, or make it blank and see all previous statistics

`Web.DBStatsSubmitDV.CanShowPreviousStats` = If yes, previous statistics stored in system tables will be shown

`Web.DBStatsSubmitDV.PreviousStatsTime` = Point In Time for Previous Statistics or leave blank to get all Previous Statistics

`Web.DBStatsSubmitDV.ShowHistoricalStats` = Show Previous Statistics

`Web.DBStatsSubmitDV.ShowPreviousStats` = Show Previous Statistics

`Web.InternalTools.InfoPages.DatabaseStorage.SelectStorageSetToDisplay` = Select Storage Set To Display

`Web.InternalTools.InfoPages.ServerPerformance.ClearResults` = Clear Results

`Web.InternalTools.InfoPages.ServerPerformance.ExportRaw` = Export Raw Data to CSV

`Web.InternalTools.InfoPages.ServerPerformance.ExportSummary` = Export Summary to CSV

`Web.InternalTools.InfoPages.ServerPerformance.Title` = Server Performance

`Web.InternalTools.InfoPages.UpdateStatistics` = Update Statistics

`Web.InternalTools.InfoPages.UpdateStatistics.AvailableUpdateStatisticsRuns` = Available Update Statistics Runs

`Web.InternalTools.InfoPages.UpdateStatistics.BatchParams` = Update Statistics Batch Job Parameters

`Web.InternalTools.InfoPages.UpdateStatistics.Incremental` = Incremental

`Web.InternalTools.InfoPages.UpdateStatistics.IncrementalUpdateStats` = Incremental Update Statistics

`Web.InternalTools.WorkQueueLv.DownloadHistory` = Download History

`Web.Profiler.ClickOnStackNameAndPressThisButtonToViewStackTree` = Click on Stack Name and Press this Button to View Stack Tree

Modify the following display keys in `display.properties`

`Java.Encryption.ConfigError.DuplicateEncryptionId` = Multiple IEncryption plugins configured with the encryption id "{0}": {1} and {2}.

Remove the following display keys in `display.properties`

`Java.Dev.Mode.Label` = [DEV mode - {0}]

`Java.ServerMode.DevModeWarning` = The server is in "development" mode. This mode should never be used for a production server.

Modify the following in `gosu.display.properties`

`MSG_BAD_NAMESPACE` = {0} is not a valid namespace.

`MSG_ABSTRACT_METHOD_CANNOT_BE_ACCESSED_DIRECTLY` = Abstract method {0} cannot be accessed directly

`MSG_FUNCTION_NOT_OVERRIDE_PROPERTY` = Function {0} does not override anything. Perhaps you mean to override a property {1}?

`MSG_RELATIONAL_OPERATOR_CANNOT_BE_APPLIED_TO_TYPE` = Relational operator {0} cannot be applied to type {1}

`MSG_TYPE_HAS_XXX_ACCESS` = Type {0} has {1} access

MSG_CANNOT_EXTEND_RAW_GENERIC_TYPE = The generic type {0} requires type argument[s]

MSG_SUPER_CLASS_METHOD_NAME_SAME_AS_SUBCLASS = The super class method {0} conflicts with this class' internal constructor name. Please change either the super class method name or this class name.

MSG_CALL_TO_SUPER_THIS_MUST_BE_FIRST = Call to {0}() must be the first statement in the constructor

Add the following to studio.display.properties

ExportTypeListAction.Name = {0} to {1}

DisplayNameColumnsEditorPanel.Use_EntityName = Use Entity Name?

Modify the following to studio.display.properties

DisplayKeyController.Editing_display_keys_will_cop = Editing display keys will create display.properties files for all locales in the current module. Continue?

NewWorkflowssubtypeMetadataDialog.caption = New Workflow Subtype Metadata

Add the following to BatchProcessType.ttx

```
<typecode code="ReviewSync" desc="Transmits completed reviews to ContactCenter."
  name="ClaimCenter (SPM) Completed Review Sync">
  <category code="UIRunnable" typelist="BatchProcessTypeUsage"/>
  <category code="Schedulable" typelist="BatchProcessTypeUsage"/>
  <category code="APIRunnable" typelist="BatchProcessTypeUsage"/>
</typecode>
```

Remove the following from BatchProcessType.ttx:

```
typecode code="BulkInvoiceSubmission"
```

```
----> removed <category code="Schedulable" typelist="BatchProcessTypeUsage"/>
```

Add the following to OfficialID.ttx

```
<typecode code="NCCIintrastate" desc="NCCI Intrastate ID" name="NCCI Intrastate ID"/>
```

Display Keys for ContactCenter

The following section affects ContactCenter customers who have a localized build. You must either add, modify, or delete the following display keys in your customer configuration file, specifically in your locale folder. Make your changes for all the display keys in the applicable file (for example, display.properties, or studio.display.properties files.)

WARNING If you fail to make these changes to your configuration, then the application will not start.

Add the following display keys in display.properties

Java.Database.NoCollationDefined = No collation name found in collations.xml for locale {0}

Java.Database.NoSortCollationDefinedForOracle = A specific Oracle collation for database sorting was not configured in collations.xml for the default application locale {0}

Java.Database.NoSortCollationDefinedForOracleDefaultUsed = Collation BINARY_CI will be used instead.

Java.Database.UnsupportedDatabaseVersionAllowedInDevMode = Unsupported database version allowed because server is running in development mode

Java.Database.UnsupportedJDBCDriverAllowedInDevMode = Unsupported JDBC driver allowed because server is running in development mode

Java.Entity.Error.IdentifierWithDisplayName = '{0}\:{1} - {2}'

Java.Entity.Error.IdentifierWithoutDisplayName = '{0}\:{1}'

Java.Error.UpdateStatsDVHelper = Update Statistics batch process is already running

Java.InternalTools.InfoPages.DataDistribution.ComparisonRequiresTwoOrMore = To download a comparison of data distributions, you must select two or more data distributions

Java.ServerMode.Label.Dev = [DEV mode - {0}]

Java.ServerMode.Label.Test = [TEST mode - {0}]

Java.ServerMode.Warning.Dev = The server is in "development" mode. This mode should never be used for a production server.

Java.ServerMode.Warning.Test = The server is in "test" mode. This mode should never be used for a production server.

Web.Admin.WorkflowDetail.Parent = Parent Workflow

Web.Admin.WorkflowDetail.SubFlows = Children Workflows

Web.Admin.Workflows.Children = Children

Web.Admin.Workflows.Parent = Parent

Web.DBStatsSubmitDV.CanChoosePointInTime = A specific point in time can be selected to display statistics that were current at that time, or make it blank and see all previous statistics

Web.DBStatsSubmitDV.CanShowPreviousStats = If yes, previous statistics stored in system tables will be shown

Web.DBStatsSubmitDV.PreviousStatsTime = Point In Time for Previous Statistics or leave blank to get all Previous Statistics

Web.DBStatsSubmitDV.ShowHistoricalStats = Show Previous Statistics

Web.DBStatsSubmitDV.ShowPreviousStats = Show Previous Statistics

Web.InternalTools.InfoPages.DatabaseStorage.SelectStorageSetToDisplay = Select Storage Set To Display

Web.InternalTools.InfoPages.ServerPerformance.ClearResults = Clear Results

Web.InternalTools.InfoPages.ServerPerformance.ExportRaw = Export Raw Data to CSV

Web.InternalTools.InfoPages.ServerPerformance.ExportSummary = Export Summary to CSV

Web.InternalTools.InfoPages.ServerPerformance.Title = Server Performance

Web.InternalTools.InfoPages.UpdateStatistics = Update Statistics

Web.InternalTools.InfoPages.UpdateStatistics.AvailableUpdateStatisticsRuns = Available Update Statistics Runs

Web.InternalTools.InfoPages.UpdateStatistics.BatchParams = Update Statistics Batch Job Parameters

Web.InternalTools.InfoPages.UpdateStatistics.Incremental = Incremental

Web.InternalTools.InfoPages.UpdateStatistics.IncrementalUpdateStats = Incremental Update Statistics

Web.InternalTools.WorkQueueLv.DownloadHistory = Download History

Web.Profiler.ClickOnStackNameAndPressThisButtonToViewStackTree = Click on Stack Name and Press this Button to View Stack Tree

Modify the following display keys in display.properties

Java.Encryption.ConfigError.DuplicateEncryptionId = Multiple IEncryption plugins configured with the encryption id "{0}": {1} and {2}.

Remove the following display keys in display.properties

Java.Dev.Mode.Label = [DEV mode - {0}]

Java.ServerMode.DevModeWarning = The server is in "development" mode. This mode should never be used for a production server.

Modify the following in gosu.display.properties

MSG_BAD_NAMESPACE = {0} is not a valid namespace.

MSG_ABSTRACT_METHOD_CANNOT_BE_ACCESSED_DIRECTLY = Abstract method {0} cannot be accessed directly

MSG_FUNCTION_NOT_OVERRIDE_PROPERTY = Function {0} does not override anything. Perhaps you mean to override a property {1}?

MSG_RELATIONAL_OPERATOR_CANNOT_BE_APPLIED_TO_TYPE = Relational operator {0} cannot be applied to type {1}

MSG_TYPE_HAS_XXX_ACCESS = Type {0} has {1} access

MSG_CANNOT_EXTEND_RAW_GENERIC_TYPE = The generic type {0} requires type argument[s]

MSG_SUPER_CLASS_METHOD_NAME_SAME_AS_SUBCLASS = The super class method {0} conflicts with this class' internal constructor name. Please change either the super class method name or this class name.

MSG_CALL_TO_SUPER_THIS_MUST_BE_FIRST = Call to {0}() must be the first statement in the constructor

Add the following to studio.display.properties

ExportTypeListAction.Name = {0} to {1}

DisplayNameColumnsEditorPanel.Use_EntityName = Use Entity Name?

Modify the following to studio.display.properties

DisplayKeyController.Editing_display_keys_will_cop = Editing display keys will create display.properties files for all locales in the current module. Continue?

NewWorkflowsubtypeMetadataDialog.caption = New Workflow Subtype Metadata

Known Issues and Limitations

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

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.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the ClaimCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the Default_Server_stderr.log file will show an interruption in the “Applying upgrade step ### of ####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

Must regenerate database statistics after large batch processing of addresses (CLM-13963)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in `filename`.

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()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, then continue to use the deprecated methods.

Link issues between a new contact from the New Check wizard and ContactCenter (CLM-12172)

Issue: If you try to link a contact newly created from the New Check wizard to ContactCenter, the contact details can be overridden if the contact already exists in ContactCenter. This occurs if there is a match between the two integrated systems. Rather than creating a new contact, the existing matching contact information in ContactCenter overwrites the contact's information in ClaimCenter.

Workaround: Create the contact first in ClaimCenter (not in the New Check wizard), then if you want, link it to ContactCenter. Finally, use the New Check wizard to create a check for that contact.

SampleAcrobat document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

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);"`

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.

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: one for pjpeg and one for normal jpeg. This causes a problem as the descriptions of both are identical. If you upload a normal 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.

License state drop-down menu in the user interface produces unexpected results (PL-9690)

Issue: If you use the `AddressAutocompleteUtil.getStates()` method to populate license states in the `FNOLVehicleIncidentPopup.pcf` file, then you might see unexpected results, such as countries or retired type-codes. This occurs when you select the license state drop down menu in the user interface. What is happening is that `AddressAutocompleteUtil.getStates` is passing `true` to the `getTypeKeys` method when it should be `false`.

Workaround: Use the following:

```
function getStates_Ext(country : typekey.Country) : typekey.State[] {
    var stateList = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = "US"; //default to US
    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
            stateList.add(eachState)
        }
    }
    return stateList.toArray() as typekey.State[]
}
```

Include this function in your PCF file. Instead of calling the `AddressAutocompleteUtil` method in the `valueRange`, call the following method instead:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

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.

Auto First and Final Claim wizard displays an exception if you change the policy currency through an independent wizard step (CLM-13498)

Issue: In the FNOL wizard, after selecting the policy and continuing in Auto First and Final mode, there are menu options on the left side of the screen to edit the details of the policy. If you edit the policy and change the currency, the new currency is not set in the NewClaimCheck object, resulting in an exception in the user interface. This occurs when you click **Add Deductible**. Note that editing the policy and changing its currency from its currency in the policy system makes the policy unverified.

Workaround: After editing the policy's currency, go back to Step 1 of the wizard, then click **Next** to return to Step 2. This updates the NewClaimCheck appropriately.

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 goes down the left side instead of across.

Workaround: Guidewire is aware of this issue.

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: Guidewire is aware of this issue.

Localized document pattern specified in the activity pattern is not passed to the activity (CLM-13670)

Issue: When ClaimCenter is localized to the Japanese language, and you create an activity using the *Gosu Sample Email Sent Record* activity pattern, the system does not select the correct language.

Workaround: Use the document search feature to find the applicable localized document template.

Integrity and consistency checks on multicurrency amounts are not complete (CLM-13243)

Issue: The integrity and consistency checks on multicurrency amounts are not yet as strict as the enforcement that exists when new transactions are created in ClaimCenter.

This is specifically for the entities: TransactionLineItem, Deduction, and CheckPortion. Any two amounts on an entity should be equal when they are in the same currency. For example, for an instance of TransactionLineItem, TransactionAmount must equal ClaimAmount if Transaction.Currency==Transaction.Claim.Currency. It is similar for TransactionAmount-

ReportingAmount and ClaimAmount-ReportingAmount. For CheckPortion, the enforcement is only performed if FixedTransactionAmount, FixedClaimAmount, and FixedReportingAmount are not null.

Workaround: For multicurrency implementations, manually ensure during the population of staging tables that the amounts on the entities TransactionLineItem, Deduction, and CheckPortion are consistent, given their currencies. For single-currency implementations, all amounts should be the same, avoiding the issue.

Exception in the Bulk Invoice screen after restoring and reopening an archived claim (CLM-174)

Issue: If you archive a claim, restore it within 60 minutes of archiving it, and then immediately attempt to view a bulk invoice involving that claim, the system may show an exception message. This is because it takes time for the global cache to be updated after the claim is archived. However, this does not cause any data loss or corruption.

Workaround: You must wait 60 minutes after archiving a claim before restoring it. The length of time to wait is determined by the GlobalCacheStaleTimeMinutes configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt to view the bulk invoice for 60 minutes. Guidewire may not fix this issue, since restoring archived claims so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

ClaimCenter login remains active in new browser window (CC-4152)

Issue: When you log into ClaimCenter in a browser window, then you close that window and open a new one, in some instances you can access ClaimCenter pages without needing to log in again.

This is an issue with some versions of Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with ClaimCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Some versions of Internet Explorer do not allow you to set this option, and enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928. However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the ClaimCenter window before closing your browser.

Optionally, you can take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer.

WARNING Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess] registry entry. Set the data type to REG_SZ, and then set the data value to yes.

Port conflict when starting ClaimCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is a port that the ClaimCenter JMX RMI adapter can use, *if enabled*. The ClaimCenter server then reports a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the ClaimCenter server again. You can also change the port that the RMI adapter uses by editing the JMXManagementPlugin through Guidewire Studio.

Guidewire Studio PCF Editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Status: Guidewire is aware of this issue.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because ClaimCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

ClaimCenter cannot load zone data with missing values (PL-9123)

Issue: Zone data is used for associating postal codes with cities and regions. You cannot load zone data that has empty fields.

Workaround: Guidewire is aware of this issue. As a temporary workaround, you can enter a character into that field, for example, a hyphen (-).

H2 development database creates LONGTEXT fields as VARCHAR (65000), instead of as CLOB as in Oracle and SQL Server (PL-9314)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database. Note that the H2 database is *only* used for testing and should never be used in production.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

ClaimCenter does not display empty report folders (PL-1281)

Issue: ClaimCenter does not display a report folder in the **Administration** → **Report Admin** page if that report folder is empty.

Workaround: Within InetSoft Enterprise Manager, insert a replet (report template) into the empty report folder. You can then mark this replet as not visible (using the InetSoft Enterprise Manager), which causes the replet to not show in ClaimCenter. However, ClaimCenter does display the parent folder.

Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled (PL-9813)

Issue: If you have the Windows Server 2003 component called IE Enhanced Security Feature installed, tabs may be unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server.

Country specific field validation does not run if the fields are not modified when a country is changed (PL-9828)

Issue: There may be an error if you change a contact field in the user interface that is validated with country specific field validators. Those specific validators do not get revalidated if the country changes and the field is not modified. This issue only occurs if you use country specific field validators.

For example, suppose that a carrier operates in two countries, the United States and Canada and wants to implement different field validation for certain data types such as Tax ID. If an adjuster changes the country on an existing contact record from the United States to Canada, without modifying the Tax ID to reflect the new format, the system allows the US formatted Tax ID to be saved. If the adjuster *edits* the Tax ID, then the system revalidates the format.

Status: Guidewire is aware of this issue.

In Japanese, the Other Coverage Info Details label displays vertically (CLM-14302)

Issue: The Other Coverage Info Details label on the Exposure Details screen displays vertically in Japanese.

Workaround: In Studio, move the Other Coverage Info list view to the page footer.

Restoring a claim fails if there is no assignment and the default owner is not in a group (CLM-13205)

Issue: A null pointer exception is thrown in the user interface if you removed the default owner from the root group, and you attempted to assign the claim.

Workaround: Put the default owner in a group, preferably the root group (default).

No warning that you will overwrite a Service Provider Review or Question Set (CLM-14365)

Issue: If you do not retire the review or question set and import the new review or question set, then there is no warning or mechanism to prevent you from overwriting the existing review or question set.

Workaround: You should create the new review or question set in a new .csv file. Also, ensure that all public IDs of all entities of the new review or question set are unique and different from the existing review or question set. Consider that any entities such as ReviewType, ReviewCategoryQuestionSet, QuestionSet, Question, QuesitonChoice, and QuestionFilter could be overwritten.

RecoverySet can have at most one uncanceled recovery consistency check error (CLM-12925)

Issue: Multiple voided recoveries contained within the same transaction create reports of database inconsistencies. This is because the SQL query on the consistency check is incorrect.

Workaround: You can safely ignore the consistency check.

User without view note permission is able to view medical note (CLM-14077)

Issue: A user without the *view note* permission is able to view medical notes on the Notes screen

Workaround: You can add a visibility constraint using permissions to the row iterator displaying notes. This will ensure that medical notes are not displayed to users without the view medical notes permission. Note that the record count will display the complete count with this workaround, even though certain rows are hidden.

Unable to generate a Claim Metrics report through the InetSoft Report Portal (CLM-13141)

Issue: The InetSoft onLoad script is designed to be run from within ClaimCenter and it should pass the report the locale. If the report is run outside of ClaimCenter, the locale needs to be set within InetSoft.

Workaround: Run reports through ClaimCenter.

New documents fail to show up in *ActivityDocumentsLV* of *ActivityDetailWorksheet* if ActiveX is disabled (CLM-12928)

Issue: If document ActiveX is disabled in the server config file, the newly created documents on an activity failed to show up in *ActivityDocumentsLV* of *ActivityDetailWorksheet.pcf* file.

Workaround: Manually link (using the **Link** button) the newly created document to the *ActivityDetailWorksheet.pcf* file.

Tax ID fields are not masked in claim snapshots that were created in earlier versions of ClaimCenter (CLM-12879)

Issue: Tax ID fields in the claim snapshot are not masked in the user interface. This happens because the claim snapshot is using an earlier version of the *ClaimSnapshotContactxxx.pcf* file which does not have the field encryption on the claim snapshot. (xxx represents various versions of the pcf file.)

Workaround: You must manually add a mask to the encrypted fields in earlier versions (such as 5.0.0) on the *ClaimSnapshotContactxxx.pcf* file.

Report folders do not display if empty (PL-1281)

Issue: Guidewire ClaimCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in ClaimCenter even if the folder is empty.

Not possible to hot-deploy on all supported application servers (PL-9955)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

Values in MIME-type drop-down menus are not localized (PL-10921)

Issue: ClaimCenter does not localize the values in the **MIME type** drop-down list properly.

Workaround: If you want to localize MIME types, you can do so by modifying the definition in *config.xml*. Currently, ClaimCenter does not support MIME type definitions in multiple languages.

Finder in annotation causes throws error (PL-10812)

Issue: Studio throws error and fails to do syntax error highlight, code completion, and similar functions, if there is a finder in an annotation. This occurs even if you are connected to the server.

Workaround: Remove the finder from the annotation.

Navigating between pages in wizard loses data (PL-10920)

Issue: In wizard popups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Studio does not properly report verification error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Double quotes in PATH environment variable causes problems (PL-10981)

Issue: ClaimCenter does not run if there are double quotes in the PATH environment variable.

Workaround: Check the PATH variable and remove any double quotes.

Studio Debugger does not handle rule breakpoints properly (PL-10553)

Issue: If you set a breakpoint in a Gosu rule and attempt to step through the code using the Studio debugger, the debugger does not correctly display the code on the breakpoint step.

Status: Guidewire is aware of this issue.

Inability to access typecodes that contain a dash (PL-10532)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

Method *assignByUserAttributes* does not distinguish between attributes (PL-10607)

Issue: Method *assignByUserAttributes* does not distinguish between different attributes.

Workaround: Ensure that you thoroughly test all assignment scenarios to ensure that assignment happens correctly.

Cache summary statistics rely on server ID being unique (PL-11239)

Issue: If you use the same server ID for multiple servers in a cluster, it can create the following problems:

- Different servers in a cluster deleting or changing each others results
- Deadlocks at the database level

Status: Guidewire is aware of this issue.

InetSoft Exception *Failed Login Exception: Bad user name anonymous* (PL-10148)

Issue: InetSoft generates a *Failed Login Exception: Bad user name anonymous* exception on server start-up. This occurs if you set the InetSoft logging level to *fine*.

Workaround: This error does not cause any issues. You can safely ignore the exception.

Report Still Visible After Permissions Disabled For Parent Folder (PL-10353)

Issue: An InetSoft report that is included in a permission set is still visible in the report tree even after its parent has been removed from the permission set.

Workaround: You must remove individual reports from permission sets instead of maintaining the reports at the report folder level.

ToolBarButton Widget Does Not Render Correctly (PL-10742)

Issue: ClaimCenter incorrectly renders a ToolBarButton widget if the associated AddMenuItemIterator is empty.

Workaround: Modify your configuration to disable the menu button if the button contains no menu item.

Must Name Row Iterator Element with Capital Letter (PL-10795)

Issue: The Document search results page (DocumentLV.pcf) throws a *NullPointerException* if the row iterator element name is *document* (with a lower case d).

Workaround: You must use *Document* (with a capital D) instead.

Large Number of Classes in `plugins\shared\lib` Causes Studio to Fail to Start (PL-11026)

Issue: If you attempt to load multiple JAR files that contain many thousands of classes, Studio fails to start and throws an `OutOfMemory` error.

Workaround: Increase `MaxPermSize` to something over 128m (the default) so that Studio has enough memory to build the type information database.

ANT Command *build-war* Hangs Indefinitely (PL-11045)

Issue: It is possible for the `build-war` ANT command to hang indefinitely. This issue is only reproducible in certain environments.

Workaround: Set the `-noinput` parameter before calling the `build-war` ANT command. For example, instead of `gwcc.bat build-war`, do the following:

```
gwcc.bat -noinput build-war
```

Missing Display Keys Cause Server to Fail to Start (PL-11764)

Issue: Missing display keys defined in the application default locale can cause the server to fail to start.

Workaround: Within Guidewire Studio, run the **Verify Resources** utility before attempting to start the application server. ClaimCenter bases the type system for display keys on those defined in the default application locale. If a display key in the default application locale is missing and a rule references the missing display key, then ClaimCenter cannot compile the rule. Running the resource verification utility can catch these types of missing display keys.

ClaimCenter Not Generating *soap.local.entity* Correctly (PL-7560)

Issue: ClaimCenter is not generating the `soap.local.entity` for a class used by web services.

Workaround: To ensure that ClaimCenter generates the `soap.local.entity` correctly for `FieldValue` classes (which are Java types), subtype `FieldValue` into `GosuFieldValue`, then further subtype that class.

Unsuccessfully stopped/voided payments should not be recodeable (CLM-14207)

Issue: On the **Payment Detail** screen (`ClaimFinancialsTransactionsDetail.pcf`), the **Recode** button is available for payments that have been *uncanceled*, even though only the onset payment created during uncancellation should actually be re-codeable.

Workaround: In the *available* attribute of the `TransactionDetailToolBarButtonSet_RecodeButton` element of the `TransactionDetailToolBarButtonSet.payment.pcf` file, append this: and `Payment.Onset==null`.

Deleting an user who owned an archive claim and running the archive worker item throws a `DBDuplicateKeyException` (CLM-12868)

Issue: If a user is deleted after a claim has been archived and restored, then the archive worker item throws a `DBDuplicateKeyException` exception.

Workaround: Set the user who owns an archive claim to inactive instead of deleting them.

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.

Hidden reports should not be visible in Report Admin tree (PL-12313)

Issue: If you set a report's visibility to false in InetSoft, you can still see the report in the ClaimCenter **Admin Report** tree. The path is **Administration** tab → **Report Admin**. However, these reports still remain hidden to any report users from both the InetSoft Portal and in ClaimCenter.

Status: Guidewire is aware of this issue.

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"/>
```

Convert this to a generic Input widget, without the format attribute. Next add a custom entity display name for the `ExchangeRate` entity in Studio (path is **configuration** → **Entity Names**). You can customize how you would like the exchange rate to show.

User is allowed to delete a Queue that has open Activities (CLM-14576)

Issue: If you delete a queue that has activities assigned to it, the activities cannot be reassigned to a user. The Group Admin was allowed to delete a queue that had an open activity. Since the queue is now retired, the activity is now visible only from the Workplan screen. The activity can no longer be taken from the queue, since the queue is no longer visible. This means that the claim can never be closed since it has an open activity that cannot be skipped or completed.

Workaround: To prevent accidental deletion of a queue to which one or more open activities are assigned, use a toolbar flag to control availability of the **Remove** button in GroupQueues detail view. The **Remove** button should be available if and only if the user has selected one or more queues that have no open activities.

Drill down to Claim Catastrophe Detail report displays "No Records Found" when there are claims (CLM-14961)

Issue: If you drill down to the **Catastrophe Financials** report located under the **Financials** folder on the **Report** tab, ClaimCenter displays an incorrect message stating there are no records when indeed, records exist.

Status: Guidewire is aware of this issue.

Drill down report from Claim Financials by Claim Owner does not include claim level transactions (CLM-14962)

Issue: If you drill down to the **Catastrophe Financials** report located under the **Financials** folder on the **Report** tab, and click **Property** to drill down to the **Catastrophe Financials by Claim Owner**, the report does not include claim-level transactions.

Status: Guidewire is aware of this issue.

The .csv file is incorrectly configured in mimetype mapping (PL-12841)

Issue: In ClaimCenter, the .csv file is in a different format than what is specified by the file extension.

Workaround: Update the mimetype in your config.xml file.

Linguistic Search Performance on Oracle is Slow (PL-12143)

Issue: ClaimCenter uses an Oracle Java Stored Function to perform linguistic searches. Such search queries can perform poorly. ClaimCenter invokes the stored function on Oracle if either one of the following is true:

- The primary locale language is German or Japanese
- The `LinguisticSearchCollation` strength is set to `primary` in the default locale's `GWLocale` element in file `localization.xml`.

This is an issue for any linguistic search in which you set the `LinguisticSearchCollation` strength to `primary`. For example, if you are working in a language that commonly uses accents (such as French or Spanish), and you wish to perform accent-insensitive searches, then you typically set the `LinguisticSearchCollation` strength to `primary` for that language. If you do so, it is possible to experience slow performance on an Oracle database as you perform linguistic searches.

Workaround: Guidewire recommends that you set the `LinguisticSearchCollation` strength to `secondary` if applicable. (This is not applicable for Japanese and German.)

ABReviewSummary WSDL is not refreshing (CLM-14975)

Issue: The issue occurs in a ClaimCenter / ContactCenter integrated environment when a service provider review is completed and the `ReviewSync` distributed work queue runs to push the new review to ContactCenter. The error occurs due to an out of date WSDL for the `ABReviewSummary` web service.

Workaround: You must refresh the `ABReviewSummary` web service. Do the following:

1. Start up ContactCenter.
2. Open ClaimCenter Studio and open `ABReviewSummary` (configuration → **WebServices**).
3. Click **Refresh** (located in the upper right corner of the `WebService Details` view) and save.
- 5) Rebuild and redeploy the ClaimCenter WAR file.

chapter 36

Guidewire ClaimCenter 6.0.3 Release Notes

Release 6.0.3.17

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Insurance Suite, Guidewire ContactCenter, Guidewire Studio, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries. All other product names are trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

- This release of Guidewire ClaimCenter is 6.0.3.17.
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter
- Prior ClaimCenter release notes for any versions that have been skipped

The following issues pertain to installing this release.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-798-ERX-0000BF00100001F-F2AFC9DD34ED

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Issues and Major Changes

This section contains issues or changes that may affect your installation.

Archive Functionality

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 implementing archiving to help your company with this analysis.

Upgrade Issues

This section describes changes to the ClaimCenter base configuration that may cause upgrade issues.

Document Assistant ActiveX

Added a second version of the Guidewire Document Assistant ActiveX control that Guidewire provides to manage many of the functions related to document manipulation. The default version uses a non-configurable

whitelist of file types that the Document Assistant control will open. It uses the application associated by the operating system with that file type. If the file type is not on the whitelist, Guidewire Document Assistant will not open the file. The new blacklist version provides a list of known potentially dangerous file types that Guidewire Document Assistant prevents from opening under any circumstances. The blacklist version also includes a configurable whitelist of file types to open. The Document Assistant will open any file type on the whitelist, unless it is also listed on the blacklist. If a file type is on neither list, the Document Assistant prompts the user to save the file.

You can select which version of the Document Assistant to use. If you select the blacklist version, you can configure the whitelist of allowed file types. For information on the default allowed and unallowed file types and how to configure either version of the Guidewire Document Assistant ActiveX control, see the System Administration Guide, topic *Configuring Guidewire Document Assistant*.

Display Key Changes

If you have implemented the localized modules, then you must modify the display keys in your application. Refer to *this file* for details on the changes.

Subrogation cookbook implementation can cause upgrade issues (CC-53120)

Issue: ClaimCenter 4.0 customers that implemented the *Subrogation Cookbook*, article ID # 286 on the Guidewire *Customer Service Center Portal*, may run into difficulties when upgrading unless steps are taken prior to upgrade. The cookbook provides to ClaimCenter 4.0 customers similar subrogation functionality to what was released in ClaimCenter 5.0. If you implemented this cookbook without first modifying the entity names listed in the cookbook, then you will not be able to upgrade to ClaimCenter 5.0 or 6.0 without first resolving a naming conflict.

One way to determine if this issue applies is if any of the following entities are found in the `extensions.xml` file: `SubroAdverseParty`, `StatuteLimitationsLine` or `SubroPaymentSchedule`. If you find those entities but they have a slightly different name, such as `subroadverseparty_ext`, then no conflict should occur.

Workaround: If one or more of those entities are found in the `extensions.xml` file, then you have two options.

- Rename the entities so that no conflict exists. For example, change `SubroAdverseParty` to `subroadverseparty_ext`. Migrate the data and update all related `.pcf` files and Gosu code, *or*
- Migrate the data from the current entities to the default configuration entities.

For detailed information on these options, refer to article ID #643 on the Guidewire Customer Service Center Portal.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Rules Changes
- Improvements to Upgrade Diff Report
- Improvements and General Issues
- Known Issues and Limitations

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 6.0.2 to 6.0.3

- 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/pl` directory, [click here](#).

Rules Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 6.0.2 to 6.0.3

- To view a report of the changes in the base rules in the `modules/cc` directory, [click here](#).

Improvements to Upgrade Diff Report

With previous releases, Guidewire provided a static report in the release notes detailing certain differences between the current release and the prior minor release. This report described changes in display keys, entities, typelists, and the Gosu API. Guidewire has improved this report to dynamically conform to each customer's particular upgrade path. In addition, the report is regularly updated as the tools for generating it are enhanced to provide more information.

Because the new report is tailored to your particular upgrade requirements, it is no longer included with the general release notes. To obtain your custom Upgrade Diff Report, contact your Guidewire representative. In the future, this report will be available on the Guidewire Resource Center web portal, allowing you to view the latest version right when you need it.

Improvements and General Issues

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

ID	Description
Administration	
CLM-12989	Corrected an issue where the Team User Claims page generated a query that had duplicate predicates.
Claim	
CLM-13154	Corrected an issue where you might get a <code>NullPointerException</code> if you partially completed the New Claim wizard, navigated away and returned to it from the Unsaved Work menu.
CLM-15330	Fixed an issue where the new ICD reference data now has the correct suffix. For example, a code will have 100.0 instead of 100 and 34. instead of 34.
CLM-15497	<p>Corrected a run time exception that appeared when viewing FNOL snapshots from a minor release after viewing a snapshot from a major release (x.0.0 or 3.1.0). The fix modified type checking on the snapshot pages. It is possible the fix may interact with customer-written snapshot-related code to cause typecast exceptions when viewing snapshots.</p> <p>Perform the following test steps to determine if this issue affects your installation:</p> <ul style="list-style-type: none"> • Identify the ClaimCenter releases that you have used in production and the approximate dates they went live. • Based on those dates, find claims that were created with each of those versions. Guidewire suggests testing one claim created with each major release that has been in production, newest to oldest (6.x.x, 5.x.x, and so forth). • For each claim, go to the FNOL snapshot page and click each card in the screen (Loss Details, Parties Involved, Policy, Exposures, Notes, Documents, Additional Fields). Contact Guidewire support for assistance if you see errors such as: <code>EvaluationException: Cannot coerce snapshot.v603.Claim to snapshot.v600.Claim</code>.
Command-line Tools	
CLM-14901	Fixed an issue where there were integrity check errors on data loaded from staging tables. This was due to orphaned <code>PolicyPeriod</code> records.
Contacts	
CLM-14811	Fixed an issue where if you removed the last role from a claim contact listed on the Parties Involved screen, ClaimCenter would instantly remove that party from the list of contacts and shift the edit view to a different party.

ID	Description
Core-Platform	
PL-11189	Guidewire has determined that an intermittent verification error on <code>autoCompleteArgIds</code> occurs due to a JVM bug. You can safely ignore this error. Or, if you choose, you can upgrade your JVM to 1.6.0_21 or later, which removes the cause of the error message.
PL-11310	Corrected an issue in which, if you were defining the <code>sortOrder</code> for an element related to a query-backed <code>ListView</code> , Studio did not report an error if you chose an element for sorting that could not be used for sorting the backing query.
PL-11572	Corrected an issue that caused a <code>NullPointerException</code> if you canceled out of a page that contained unsaved work and the page contained data that had been deleted in the database.
PL-11691	Added a tool that generates a report listing the Gosu rules defined in the Rules editor in Guidewire Studio. To generate the report, type <code>gwcc regen-rulereport</code> in the application bin directory. See the System Administration Guide for more information.
PL-12314	Modified the data model and added a new element <code>remove-index</code> to the data model elements. You can use this element to remove an index if it exists purely for performance. Such indexes are either a) non-unique, or b) unique but contain an ID column. (Guidewire performs metadata validation to ensure that the element removes only those indexes that fall into one of these categories.) To modify a database index, you can remove it using this element, then add a new index with a different name and with your characteristics.
PL-12402	Modified the <code>VenueType</code> typelist to make it non-final.
PL-12500	Corrected an issue in which the Data Dictionary did not display certain virtual properties on an entity if the virtual properties were virtual property fields on a delegate implemented by the entity.
PL-13076	Added a tool that generates a report listing the Gosu rules defined in the Rules editor in Guidewire Studio. To generate the report, type <code>gwcc rulereport</code> in the application bin directory. See the System Administration Guide for more information.
PL-13430	Corrected an issue associated with localizing a column on a custom entity that created a <code>gw.xml.XMLParseException: Could not parse XML file temp.dti</code> exception. The exception occurred if you added a <code><localization></code> element to <code><column></code> and used an underscore in the column or entity name.
Documents	
CLM-12928	Corrected an issue where if ActiveX was disabled, a new document failed to show up in the <code>ActivityDocumentsLV</code> of the <code>ActivityDetailWorksheet</code> .
CLM-14671	Corrected an issue where an error message needed to be localized.
Financials	
CLM-12864	Fixed a regression in 6.0.2 whereby the Change Recurrence Schedule page for a recurring check might incorrectly report that the recurrence exceeded available reserves when editing the amount of a check, because it was counting both the previous and new value of the edited amount.
CLM-12964	Corrected an issue where users saw an error message if they deleted a denied check with offset reserves.
CLM-12985	Fixed an issue where the system threw a <code>NullPointerException</code> in the user interface if the deduction's <code>TransactionAmount</code> and <code>ClaimAmount</code> entities are both null.
CLM-13136	Fixed an issue where calling the method <code>BulkInvoiceItem.setNonEroding()</code> more than once before a commit would not work, and could leave T-accounts in an incorrect state.
CLM-13498	Fixed an issue in the FNOL wizard, after selecting the policy and continuing in the Auto First and Final mode, if you edited the policy and changed the currency, the new currency was not set in the <code>NewClaimCheck</code> object, resulting in an exception in the user interface.
CLM-14606	Corrected an issue on the Check Transfer screen related to adding new required fields to the page. If the first attempt to transfer the check had missing required fields, subsequent attempts to fix the problem and complete the operation would still display an error. The Transfer button now properly rolls back its work if screen validation fails, so another attempt to transfer will be successful.
CLM-14749	Previously, if you edited a recurring check, deductions on subsequent checks in the recurrence would not be updated. Now, if you select Edit Recurrence from a particular check in the recurrence, deductions on that check and all later checks in the recurrence will be recalculated.
CLM-15002	Added additional detail to a message when a stack trace error occurred if the <code>Check.ScheduledSendDate</code> field was modified anywhere but in the <code>PreSetup</code> rules or through Gosu run from the user interface. This is considered a configuration error.

ID	Description
CLM-15184	Removed some display keys which held sentence fragments that were then included in other display keys. This made translation very difficult, since the context of the fragments was not fully apparent and could change the translation depending on how they were used.
CLM-15236	Fixed an issue during denial of a check or payment. If the check had a final payment which had closed it's exposure or claim, but the exposure or claim was now open at the time of denial, an error would occur when it tried to reopen the claim or exposure. The deny method now checks if the exposure or claim is already open before attempting to reopen it.
CLM-15310	Added a verification during the save of a new transaction to ensure that the system updates the T-accounts. This prevents the improper creation of a transaction which would result in its value not being reflected in financials calculations. T-accounts are only updated when transactions are created through the supported methods described in the documentation.
CLM-15358	Now a meaningful error message displays if a check delete fails. Previously an extra error displayed when trying to delete the check when you do not have permission, and the error's text was "{0}" or "null".
Integration	
CLM-15085	<p>Corrected an issue where the policy retrieval was incorrectly using the policy effective date rather than the associated claim's loss date when getting the full policy information. The fix included changing the API functions <code>retrievePolicyFromPolicySummary()</code> and <code>retrievePolicyFromPolicy()</code> in <code>PolicySearchPCPlugin.gs</code>. These functions use, respectively, the loss date set in the policy summary or the loss date in the associated claim.</p> <p>In addition, the policy search adapter was modified to throw an exception if the loss date is null. This affects customer-written code using the adapter that does not provide a loss date.</p>
PL-12964	<p>Added a second version of the Document Assistant ActiveX control that Guidewire provides to manage many of the functions related to document manipulation. The default version uses a whitelist of file types that the Document Assistant control will open and display. The blacklist version provides a list of known potentially dangerous file types that it prevents from opening under any circumstances.</p> <p>For information on how to select the appropriate version of the Document Assistant, and for information on how to install and configure the Guidewire Document Assistant ActiveX control, see the System Administration Guide, topic <i>Configuring Guidewire Document Assistant</i>.</p> <p>WARNING The Guidewire Document Assistant, like all ActiveX controls, has known security vulnerabilities.</p>
Manageability-Platform	
PL-9368	Added a Download Comparison Zip File button to the Server Tools Info Pages → Data Distribution Information page if you select multiple data distributions. This provides the ability to merge multiple data distribution reports. This change also adds an additional column on the distribution reports index page that shows the origin distribution number
PL-10943	Modified the Server Tools Info Pages → Load Integrity Checks so that it no longer downloads duplicate entries. The process now respects the status of the Allow Non Admin References : drop-down option.
PL-11812	Added a new <i>action</i> attribute to the <code><tablestatistics></code> element in <code>config.xml</code> . By default, ClaimCenter on Oracle does not generate statistics on any table used for processing work items. ClaimCenter deletes any existing statistics on these tables whenever ClaimCenter updates statistics. You can override this behavior using the action attribute of the <code><tablestatistics></code> element. See the System Administration Guide, <i>Configuring Database Statistics</i> , for details.
PL-12143	Corrected an issue that caused poor performance on an Oracle database if you performed a linguistic search in which the Search strength was set to primary or the default application locale was Japanese or German.
PL-12806	Modified the Administration → Internal Tools → Database Table Info page so that it now includes database partition information.
PL-12854	Modified the ClaimCenter upgrader so that it now ignores the status of database constraints created by Oracle supplemental logging.
PL-12889	Corrected an issue in which the Work Queue Info page did not include the Throughout column in a download to a CSV file. This occurred primarily if there was a big disparity in the dates or the worker completed immediately.
PL-13048	Modified the Server Tools Info pages → Data Distribution page to include information on the number of updated rows and the total number of updates in HTML tables.

ID	Description
PL-13461	Modified the Server Tools Info pages → Data Distribution Information download logic so that it no longer generates data on all assignables by default. Formerly, a data distribution download generated large amounts of data on all assignables, even if you did not select the Assignable by date option. Now, you must select that option to view this information.
PL-13549	Corrected an issue that caused an <code>ArrayIndexOutOfBoundsException</code> if you performed a Server Tools Info pages → Data Distribution query on a very large database.
Policy	
CLM-13121	Increased the scriptability of the <code>Claim.Policy</code> property so that it can be set from the user interface.
Printing	
CLM-14936	Fixed an issue where if you tried to print a Time Loss page that contained a list view with column headers but with no content (no data rows, only column headers) you would see a PDF generation exception.
Reporting	
CLM-12881	Added localization for the remaining labels on the InetSoft report: Claim Overall Avg.
CLM-12888	Corrected an issue where the number format in the InetSoft report: Claim Health Metrics was not in the default format with the language specified in the <code>sree.properties</code> file. Now it displays the currency amount based on the users language and country.
CLM-12889	Corrected an issue with certain InetSoft reports (such as Time to first Loss Payments, Claim Overall Avg, Claim by Group and Tier and so forth) where a claim's Paid Loss Costs as a percentage of Total Paid equaled N/A. It was treated as 0 and therefore was counted towards the calculation for average. It should have been null. This was only seen on the Oracle database.
CLM-12930	Corrected an issue where the InetSoft report calculation for Catastrophe Financials: Average Loss Amount was different than the current Catastrophe Financials report. Now the Average Loss Amount is the same for both reports.
CLM-12948	Corrected an issue in InetSoft reports where the word Sum was not localized in the Open Claim Financials Dashboard report and Past 30 Day Financials reports.
CLM-12975	Fixed an issue with InetSoft reports where the text on the pie chart in the Claims by Catastrophe Overview Report was not localized in Japanese.
CLM-13052	Corrected an issue with InetSoft reports where the first row was a comma on multiple claim and financial reports. This only occurred if InetSoft was running on an Oracle server.
CLM-13066	Corrected an issue with certain InetSoft reports (such as Paid Loss Costs as Percentage of Total Paid) where a claim's Paid Loss Costs as a percentage of Total Paid equaled N/A. It was treated as 0 and therefore was counted towards the calculation for average. It should have been null. This was only seen on the Oracle database.
CLM-13100	Corrected an issue with InetSoft reporting where the Drill Down Claim Health Metrics reports from a pie or Bar chart did not have the column headers translated into Japanese.
CLM-13112	Corrected an issue with InetSoft reports. The lines of business on the pie chart of Open Claims report were not localized.
CLM-13115	Corrected an issue where the workers' comp medical in the Open Claims Financials dashboard did not have the drill down link to the Transaction Detail on Coverage.
CLM-13128	Corrected an issue with InetSoft reports where the Past 30 day Claim Dashboard's Coverage drill down was empty even though there were existing records. The fix included rebuilding the table.
CLM-13423	Corrected an InetSoft Claims by Catastrophe overview report where the Claim Cost Paid pie chart contained a different precision display in the Catastrophe by Loss Type pie chart compared to the other types of pie charts.
CLM-13478	Fixed an issue with InetSoft reports where there were records in SIU - Matching Payment Address report, but the system displayed the message, No Records Found at the end of the record.
CLM-13562	Fixed a formatting issue with InetSoft reporting where the Claim Injury Detail report would return two pages of records, but the second page was blank.
CLM-13583	The new charting engine in InetSoft has changed the behavior of the tool tip so that it now displays an integer. For example if the number on the pie chart was 5.0, then the tool tip for it now shows 5.
CLM-13621	Corrected an issue in the InetSoft reports where the Loss State was truncated in the Claim Loss Details Report.

ID	Description
CLM-14698	Corrected an error in the sample Oracle definition file (location <code>cc → conf → oracle → datasource.xml</code>) where <code>server/dbuser</code> was incorrect. Now it is <code>dbuser/dbpassword</code> .
CLM-14882	Corrected an issue in reporting where there was more than one logical model in binding. This caused performance issues.
CLM-15388	Fixed an issue with InetSoft reports where the Average Duration columns in the Claims First Pay Productivity report were performing a <i>sum</i> when it should have been performing an <i>average</i> .
Studio	
PL-13220	Improved Studio activation performance if using SCM systems that maintain a local repository (for example, SVN and CVS) and Studio is configured with proper repository mask in Tools Options.
PL-13509	Modified Studio to provide ability for time-sensitive diagnostic logging in the Studio console. You can enable this functionality by creating a <code>studio.properties</code> file in Other Resources → logging and populating it with the following text: # enable studio debug level logging <code>log4j.category.Studio=DEBUG</code>
Upgrade: Configuration	
CLM-15339	There is now an upgrade trigger which moves <i>TransactionSet</i> rules to the correct folders.
CLM-15323	The Javadoc for the property <code>Claim.AllowedContactRolesForEntity</code> was incorrect. It was changed to read: Returns all allowed contact roles for a claim.
Web	
PL-12262	Corrected an issue with Date input containing Japanese characters that sometimes caused the date text to not be fully visible in the text box. This caused the user to scroll the text to see the last part of the date.
PL-12842	Improved slow performance in Internet Explorer 7 that occurred if you moused over a page with a very long ListView and a Worksheet.
PL-13043	Modified the behavior of the Eras drop-down in the Japanese Imperial Calendar widget so that it displays only the most four recent eras (Meiji, Taicho, Showa, and Hisei).

Known Issues and Limitations

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

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.

Must regenerate database statistics after large batch processing of addresses (CLM-13963)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

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()`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, then continue to use the deprecated methods.

Link issues between a new contact from the New Check wizard and ContactCenter (CLM-12172)

Issue: If you try to link a contact newly created from the New Check wizard to ContactCenter, the contact details can be overridden if the contact already exists in ContactCenter. This occurs if there is a match between the two integrated systems. Rather than creating a new contact, the existing matching contact information in ContactCenter overwrites the contact's information in ClaimCenter.

Workaround: Create the contact first in ClaimCenter (not in the New Check wizard), then if you want, link it to ContactCenter. Finally, use the New Check wizard to create a check for that contact.

SampleAcrobat document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in SampleAcrobat.pdf.descriptor and SampleAcrobat.pdf files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

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);"`

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 `checkForDuplications` in the `NewClaimWizard.pcf` file as seen in the following example:

```
function checkForDuplications() {
    if (Wizard.checkForNewDuplicateClaims()) { NewClaimDuplicationsWorksheet.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="checkForDuplications()"
    ...
```

To allow **Next** to execute, you can create a new function such as:

```
function checkForDuplicationsNoWarning() {
    if (Wizard.checkForNewDuplicateClaims()) {
        NewClaimDuplicationsWorksheet.goInWorkspace(Wizard)
    }
}
```

Have the `onExit()` function of the Main Contacts wizard step call:

```
onExit="checkForDuplicationsNoWarning()"
```

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.

Duplicate descriptions for image/jpg mime types in the configuration file (CLM-14188)

Issue: In the file `modules\pl\config\locale\en_US\display.properties`, there are two entries under MIME types for jpeg file types that are defined with the same description, JPEG Image:

- `pjpeg`
- `jpeg`

These definitions cause a problem because the descriptions of both mime types are identical. If you have the **New Document** screen open to add a new image document and you click the **Document Type** drop-down list, you see two entries for **JPEG Image**. If you choose the wrong image type and add the new document, an error occurs.

Workaround: Open Guidewire Studio and do the following:

1. Navigate in the left pane to **Configuration** → **Display Keys** → **Mimetype** → **Configuration** → **image_pjpeg**.
2. Change the **Locale: English (US)** text to **Progressive JPEG Image**.
3. Save your changes and restart ClaimCenter.
4. Now in the ClaimCenter **New Document** screen, when you click the **Document Type** drop-down list, you see two choices for these file types, **JPEG Image** and **Progressive JPEG Image**.

License state drop-down menu in the user interface produces unexpected results (PL-9690)

Issue: If you use the `AddressAutocompleteUtil.getStates()` method to populate license states in the `FNOLVehicleIncidentPopup.pcf` file, then you might see unexpected results, such as countries or retired type-codes. This occurs when you select the license state drop down menu in the user interface. What is happening is that `AddressAutocompleteUtil.getStates` is passing `true` to the `getTypeKeys` method when it should be `false`.

Workaround: Use the following:

```
function getStates_Ext(country : typekey.Country) : typekey.State[] {
    var stateList = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = "US"; //default to US
    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
            stateList.add(eachState)
        }
    }
    return stateList.toArray() as typekey.State[]
}
```

Include this function in your PCF file. Instead of calling the `AddressAutocompleteUtil` method in the `valueRange`, call the following method instead:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

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 goes down the left side instead of across.

Status: Guidewire is aware of this issue.

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.

Status: Guidewire is aware of this issue.

Localized document pattern specified in the activity pattern is not passed to the activity (CLM-13670)

Issue: When ClaimCenter is localized to the Japanese language, and you create an activity using the *Gosu Sample Email Sent Record* activity pattern, the system does not select the correct language.

Workaround: Use the document search feature to find the applicable localized document template.

Integrity and consistency checks on multicurrency amounts are not complete (CLM-13243)

Issue: The integrity and consistency checks on multicurrency amounts are not yet as strict as the enforcement that exists when new transactions are created in ClaimCenter.

This is specifically for the entities: `TransactionLineItem`, `Deduction`, and `CheckPortion`. Any two amounts on an entity should be equal when they are in the same currency. For example, for an instance of `TransactionLineItem`, `TransactionAmount` must equal `ClaimAmount` if `Transaction.Currency==Transaction.Claim.Currency`. It is similar for `TransactionAmount-ReportingAmount` and `ClaimAmount-ReportingAmount`. For `CheckPortion`, the enforcement is only performed if `FixedTransactionAmount`, `FixedClaimAmount`, and `FixedReportingAmount` are not null.

Workaround: For multicurrency implementations, manually ensure during the population of staging tables that the amounts on the entities `TransactionLineItem`, `Deduction`, and `CheckPortion` are consistent, given their currencies. For single-currency implementations, all amounts should be the same, avoiding the issue.

Exception in the Bulk Invoice screen after restoring and reopening an archived claim (CLM-174)

Issue: If you archive a claim, restore it within 60 minutes of archiving it, and then immediately attempt to view a bulk invoice involving that claim, the system may show an exception message. This is because it takes time for the global cache to be updated after the claim is archived. However, this does not cause any data loss or corruption.

Workaround: You must wait 60 minutes after archiving a claim before restoring it. The length of time to wait is determined by the `GlobalCacheStaleTimeMinutes` configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt to view the bulk invoice for 60 minutes. Guidewire may not fix this issue, since restoring archived claims so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

Guidewire Studio PCF Editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Status: Guidewire is aware of this issue.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because ClaimCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

ClaimCenter cannot load zone data with missing values (PL-9123)

Issue: Zone data is used for associating postal codes with cities and regions. You cannot load zone data that has empty fields.

Workaround: Guidewire is aware of this issue. As a temporary workaround, you can enter a character into that field, for example, a hyphen (-).

H2 development database creates LONGTEXT fields as VARCHAR (65000), instead of as CLOB as in Oracle and SQL Server (PL-9314)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database. Note that the H2 database is *only* used for testing and should never be used in production.

Workaround: Define a LONGTEXT field as a VARCHAR(. . .) with the required size, for example, as VARCHAR(120000).

ClaimCenter does not display empty report folders (PL-1281)

Issue: ClaimCenter does not display a report folder in the **Administration** → **Report Admin** page if that report folder is empty.

Workaround: Within InetSoft Enterprise Manager, insert a replet (report template) into the empty report folder. You can then mark this replet as not visible (using the InetSoft Enterprise Manager), which causes the replet to not show in ClaimCenter. However, ClaimCenter does display the parent folder.

Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled (PL-9813)

Issue: If you have the Windows Server 2003 component called IE Enhanced Security Feature installed, tabs may be unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server.

Country specific field validation does not run if the fields are not modified when a country is changed (PL-9828)

Issue: There may be an error if you change a contact field in the user interface that is validated with country specific field validators. Those specific validators do not get revalidated if the country changes and the field is not modified. This issue only occurs if you use country specific field validators.

For example, suppose that a carrier operates in two countries, the United States and Canada and wants to implement different field validation for certain data types such as Tax ID. If an adjuster changes the country on an existing contact record from the United States to Canada, without modifying the Tax ID to reflect the new format, the system allows the US formatted Tax ID to be saved. If the adjuster *edits* the Tax ID, then the system revalidates the format.

Workaround: Use the Reject method--in one of its several forms--in a validation business rule to validate critical fields. ClaimCenter runs the validation rules any time that it commits a business object that implements the Validatable delegate to the database. For this case, create a validation rule that checks whether the country field has changed without other specific fields changing, and then generate an appropriate error message.

Restoring a claim fails if there is no assignment and the default owner is not in a group (CLM-13205)

Issue: A null pointer exception is thrown in the user interface if you removed the default owner from the root group, and you attempted to assign the claim.

Workaround: Put the default owner in a group, preferably the root group (default).

No warning that you will overwrite a Service Provider Review or Question Set (CLM-14365)

Issue: If you do not retire the review or question set and import the new review or question set, then there is no warning or mechanism to prevent you from overwriting the existing review or question set.

Workaround: You should create the new review or question set in a new .csv file. Also, ensure that all public IDs of all entities of the new review or question set are unique and different from the existing review or question set. Consider that any entities such as ReviewType, ReviewCategoryQuestionSet, QuestionSet, Question, QuesitonChoice, and QuestionFilter could be overwritten.

User without view note permission is able to view medical note (CLM-14077)

Issue: A user without the *view note* permission is able to view medical notes on the **Notes** screen

Workaround: You can add a visibility constraint using permissions to the row iterator displaying notes. This will ensure that medical notes are not displayed to users without the view medical notes permission. Note that the record count will display the complete count with this workaround, even though certain rows are hidden.

Unable to generate a Claim Metrics report through the InetSoft Report Portal (CLM-13141)

Issue: The InetSoft onLoad script is designed to be run from within ClaimCenter and it should pass the report the locale. If the report is run outside of ClaimCenter, the locale needs to be set within InetSoft.

Workaround: Run reports through ClaimCenter.

Tax ID fields are not masked in claim snapshots that were created in earlier versions of ClaimCenter (CLM-12879)

Issue: Tax ID fields in the claim snapshot are not masked in the user interface. This happens because the claim snapshot is using an earlier version of the ClaimSnapshotContactxxx.pcf file which does not have the field encryption on the claim snapshot. (xxx represents various versions of the pcf file.)

Workaround: You must manually add a mask to the encrypted fields in earlier versions (such as 5.0.0) on the ClaimSnapshotContactxxx.pcf file.

Not possible to hot-deploy on all supported application servers (PL-9955)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

Values in MIME-type drop-down menus are not localized (PL-10921)

Issue: ClaimCenter does not localize the values in the **MIME type** drop-down list properly.

Workaround: If you want to localize MIME types, you can do so by modifying the definition in `config.xml`. Currently, ClaimCenter does not support MIME type definitions in multiple languages.

Navigating between steps while in a popup in a wizard may lose data (PL-10920)

Issue: If you are in a wizard and enter a popup page, then type in data but do not save, then navigate directly to a different step in the wizard by clicking that step's link on the left side, the data that was entered in the popup page and not saved may be lost. For example, this can occur if you are in the FNOL wizard on the **Exposures** step, and you add a new exposure, which takes you into a popup page. If you enter data on that **New Exposure** page, then navigate directly to a different wizard step, such as **Loss Details**, the data you entered into the **New Exposure** page will be lost.

Workaround: Ensure when entering data in a wizard that you always click **Update** (or **OK** or **Save**) to return to the parent wizard step, before navigating to a different step of the wizard.

Studio does not properly report verification error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Double quotes in PATH environment variable causes problems (PL-10981)

Issue: ClaimCenter does not run if there are double quotes in the PATH environment variable.

Workaround: Check the PATH variable and remove any double quotes.

Cache summary statistics rely on server ID being unique (PL-11239)

Issue: If you use the same server ID for multiple servers in a cluster, it can create the following problems:

- Different servers in a cluster deleting or changing each others results
- Deadlocks at the database level

Workaround: By default, the server ID defaults to the hostname of the machine running the instance. This works well in many configurations, but causes problems if multiple server instances are running on a single machine. Set the `serverid` system property in the `config.xml` file on the local server host to a unique value in the cluster. If it does not currently exist, then you need to add a `<registry>` element to the `config.xml` file on the local server host.

You can also start the server with the following JVM option: `gw.cc.serverid=????`

See *Defining the Application Server Environment* in the System Administration Guide for details on both of these options.

InetSoft Exception *Failed Login Exception: Bad user name anonymous* (PL-10148)

Issue: InetSoft generates a *Failed Login Exception: Bad user name anonymous* exception on server start-up. This occurs if you set the InetSoft logging level to *fine*.

Workaround: This error does not cause any issues. You can safely ignore the exception.

Report Still Visible After Permissions Disabled For Parent Folder (PL-10353)

Issue: An InetSoft report that is included in a permission set is still visible in the report tree even after its parent has been removed from the permission set.

Workaround: You must remove individual reports from permission sets instead of maintaining the reports at the report folder level.

ToolBarButton Widget Does Not Render Correctly (PL-10742)

Issue: ClaimCenter incorrectly renders a ToolBarButton widget if the associated AddMenuItemIterator is empty.

Workaround: Modify your configuration to disable the menu button if the button contains no menu item.

Must Name Row Iterator Element with Capital Letter (PL-10795)

Issue: The Document search results page (DocumentLV.pcf) throws a NullPointerException if the row iterator element name is *document* (with a lower case d).

Workaround: You must use *Document* (with a capital D) instead.

Large Number of Classes in plugins\shared\lib Causes Studio to Fail to Start (PL-11026)

Issue: If you attempt to load multiple JAR files that contain many thousands of classes, Studio fails to start and throws an OutOfMemory error.

Workaround: Increase MaxPermSize to something over 128m (the default) so that Studio has enough memory to build the type information database.

ANT Command *build-war* Hangs Indefinitely (PL-11045)

Issue: It is possible for the *build-war* ANT command to hang indefinitely. This issue is only reproducible in certain environments.

Workaround: Set the *-noinput* parameter before calling the *build-war* ANT command. For example, instead of *gwcc.bat build-war*, do the following:

```
gwcc.bat -noinput build-war
```

Missing Display Keys Cause Server to Fail to Start (PL-11764)

Issue: Missing display keys defined in the application default locale can cause the server to fail to start.

Workaround: Within Guidewire Studio, run the **Verify Resources** utility before attempting to start the application server. ClaimCenter bases the type system for display keys on those defined in the default application locale. If a display key in the default application locale is missing and a rule references the missing display key, then ClaimCenter cannot compile the rule. Running the resource verification utility can catch these types of missing display keys.

Deleting an user who owned an archive claim and running the archive worker item throws a DBDuplicateKeyException (CLM-12868)

Issue: If a user is deleted after a claim has been archived and restored, then the archive worker item throws a DBDuplicateKeyException exception.

Workaround: Set the user who owns an archive claim to inactive instead of deleting them.

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.

Hidden reports should not be visible in Report Admin tree (PL-12313)

Issue: If you set a report's visibility to false in InetSoft, you can still see the report in the ClaimCenter Admin Report tree. The path is Administration tab → Report Admin. However, these reports still remain hidden to any report users from both the InetSoft Portal and in ClaimCenter.

Workaround: Give the Rule Admin role (which gives permission to access the Rule Admin screen) to only those users for which seeing all reports is not an issue.

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"/>
```

Convert this to a generic Input widget, without the format attribute. Next add a custom entity display name for the ExchangeRate entity in Studio (path is configuration → Entity Names). You can customize how you would like the exchange rate to show.

Drill down to Claim Catastrophe Detail report displays "No Records Found" when there are claims (CLM-14961)

Issue: If you drill down from Claim Group to the Catastrophe Financials by Claim Owner report and then clicking on a Claim Owner to access the Claim Catastrophe Detail report, ClaimCenter displays an incorrect message stating there are no records when indeed, records exist.

Status: Guidewire is aware of this issue.

Drill down report from Catastrophe Financials by Claim Owner does not include claim level transactions (CLM-14962)

Issue: If you drill down to the Catastrophe Financials report located under the Financials folder on the Report tab, and click a Claim Group to drill down to the Catastrophe Financials by Claim Owner, the report does not include claim-level transactions.

Status: Guidewire is aware of this issue.

The .csv file is incorrectly configured in mimetype mapping (PL-12841)

Issue: In ClaimCenter, the .csv file is in a different format than what is specified by the file extension.

Workaround: Update the mimetype in your config.xml file.

Studio Rules do not use correct capitalization for root object's name (PL-10740)

Issue: Ruleset root objects are not named with first letter lower-cased.

Workaround: Rules engine will issue warnings when correct case for objects is not being used.

Countries configured in zone-config.xml still generate a warning during regen-dictionary even when zone data is loaded for all of these countries (PL-11947)

Issue: Countries configured in zone-config.xml still generate a warning during regen-dictionary even when zone data is loaded for all of these countries.

Workaround: Warning message is created in error and can safely be ignored.

The user interface cannot handle starting multiple instances of a batch process (PL-12372)

Issue: The user interface cannot handle multiple instances of a batch process.

Workaround: If multiple instances of a batch process need to be executed, they must be started from the command line. Also, you should ensure that the `BatchProcess.isExclusive()` method returns `false` to allow multiple instances to be run simultaneously.

First time you click on the arrow of the typekey input, the drop down menu will not open (PL-10134)

Issue: The drop down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting `InputHelpTextOnFocus` to `false` in the `config.xml` file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

There is a length limitation on entity localization table names (PL-13360)

Issue: There is a length limitation on entity localization table names.

Workaround: Ensure that localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

Type system refresh after PCF page title change does not update corresponding menu label (PL-13057)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

The GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

US-Locations.txt file with the US geodata from GreatData has special characters that cause validation problems with United States Postal Service (USPS) data (PL-13384)

Issue: The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can process the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

Sending email with file attachment with unicode filename is not correctly handed over to the mail server (PL-13582)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

JavaToolkit.gs has incorrectly hard coded memory which results in failed regen-java-api ant task (PL-13663)

Issue: The `JavaToolkit.gs` has hard coded memory which can result in failed `regen-java-api` ant tasks.

Workaround: Increase the size of the max heap setting on line 161 of `JavaToolkit.gs` in the Ant module. The shipped value is 512.

Copy and paste does not work from phone fields (PL-13792)

Issue: The copy and paste does not work for phone fields on a contact.

Workaround: Save the contact with the newly entered phone information before attempting to either copy or paste the value into another field.

You cannot make a field from a delegate a localized column (PL-13761)

Issue: You cannot make a field from a delegate a localized column.

Workaround: Move the column to be localized off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define an enhancement property on the delegate, that *delegates* to the appropriate column, depending on the implementing entity.

Debugger break point for the rule engine does not work if first the line is a comment or a blank line (PL-12912)

Issue: The debugger break point for the rule engine does not work if the first line is a comment or a blank line.

Workaround: Ensure that the first line is not blank and does not contain a comment. Add comments after the first line.

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.

Snapshot encryption upgrade batch process fails (CLM-15527)

Issue: The snapshot upgrade work queue (`com.guidewire.cc.system.crypt.SnapshotUpgradeWorkQueue` in the `work-queue.xml` file) throws a `ClassCastException` when it is attempting to retrieve the entity type of an object snapshot. This issue can only occur if you upgrade your encryption plug-in or change your encrypted fields, and only prevents the upgrading of existing snapshots to the new encryption configuration.

Workaround: None. It does not affect the ability to view those snapshots in ClaimCenter.

Unable to create an auto first and final claim after first attempt at check creation fails (CLM-15542)

Issue: At the end of the New Claim wizard in Auto First and Final mode, it creates a check. If this fails in a way that throws an exception, such as lack of authority to submit a check for approval, the Payment on the Check is left in an incomplete state sitting in the bundle. If you then fix the cause of the exception and click **Finish** again, it tries to commit this old check in a bad state, which results in an `IllegalStateException`.

Workaround: If you receive the message **You don't have the authority to submit this item for approval** during the New Claim wizard - Auto First and Final process, you must cancel out of the wizard, correct the authority limit issue, and then start the wizard again.

Guidewire Document Assistant ActiveX control, blacklist version, requires changes to DocumentControl.gs (PL-13501)

Issue: The CLASSID and CODEBASE values for the blacklist version of the Guidewire Document Assistant are not included in the `DocumentControl.gs` file.

Workaround: If you want to use the blacklist version of Document Assistant, replace the CLASSID and CODEBASE values in `DocumentControl.gs` with the following values:

```
CLASSID="CLSID:01A307B7-5CB5-4D91-A830-68BC53F12FD6"
```

```
CODEBASE="<%= activexCodebase %>/GuidewireDocumentAssistantConfigurable.CAB#version=2,1,0,1"
```

You can access `DocumentControl.gs` in Studio at: **configuration** → **Web Resources** → **web/templates** → **document** → **DocumentControl.gs**

For more information, see the Administration Guide topic: *Configuring Guidewire Document Assistant*.

chapter 37

Guidewire ClaimCenter 6.0.4 Release Notes

Release 6.0.4.21

Copyright © 2001-2011 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Insurance Suite, Guidewire ContactCenter, Guidewire Studio, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries. All other product names are trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 6.0.4.21.

For Standard Reporting, this release supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter
- Prior ClaimCenter release notes for any versions that have been skipped

The following issues pertain to installing this release.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-72C-ERX-0000BF00100001F-F2B025280327

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Issues and Major Changes

This section contains issues or changes that may affect your installation:

- Archive Functionality
- Change in Web Service Behavior for Committing Data to the Database (CLM-15982)
- Upgrade Issues

Archive Functionality

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.

Change in Web Service Behavior for Committing Data to the Database (CLM-15982)

The original integration design for Guidewire products focused on ensuring fail-safe integration with external systems and preventing real-time failures due to concurrent data changes. To reduce the chance of unexpected

commit errors, during API calls to published web services, the application omitted verification of whether the entity is the most recent version in the database. In other words, the server skipped the check to determine if the entity changed between (a) when the application loaded it from the application cache and (b) when the bundle commits.

This means that you may be at risk to corrupting data in your database if any of the following situations exists in your web services published from a Guidewire product:

- External systems calling web services update data in a parallelized manner. If you call the same API concurrently with multiple threads to boost overall throughput, or from multiple integration points, there is risk of corrupting data.
- Web service implementations updating entities that users can modify from the application user interface, or similarly, simultaneous changes from other web service API calls, workflows, batches, or other background processes. Because it is possible for two different threads to modify the same entity, ignoring concurrent data change exception (CDCE) errors can possibly corrupt data. Recent changes can be overwritten.

Impact

Recently, some Guidewire customers have reported issues involving the previously described scenarios. This can manifest as one or more of the following:

- The sequence generator utility returning duplicate (invalid) sequence numbers. When the application commits the bundle, the SQL row insert fails due to non-unique IDs.
- ClaimCenter can generate unexpected multiple copies of subrogation summary or claim indicator entities.
- ClaimCenter does not always accurately reflect some financial Transactions in the monetary fields on `ExposureRpt` and `ClaimRpt`. This condition is reported by database consistency checks.
- ClaimCenter provides a negative or otherwise incorrect value for `TAccount.NumContributingTxns`. This condition is reported by database consistency checks.
- ClaimCenter financials consistency checks can fail.
- ClaimCenter can experience other data loss issues.

IMPORTANT Published web service APIs that perform only read-only (get) operations are completely unaffected by this issue.

Solution

All Guidewire customers must do the following:

1. Analyze web services usage to see if the above scenarios affect your implementation.
2. Run the database consistency checker tool, also known as `dbcc checker`. This checks for certain types of potentially inconsistent data modified by web services APIs.

IMPORTANT This tool does **not** detect all possible types of corrupted data.

New Web Services Configuration Parameter (PL-14984)

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

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

If you leave this parameter set to `true`, it is possible for a bundle commit to throw a `ConcurrentDataChangeException`. This can happen if another thread or cluster node modified this entity after ClaimCenter loaded it from database into the current bundle.

If this error condition occurs, then ClaimCenter catches and rethrows the `ConcurrentDataChangeException` as `gw.api.webservice.exception.SOAPRetryableException`. This is a new exception. Guidewire strongly recommends that you update all web service clients to catch this exception and retry the SOAP API call.

Leaving `UseSafeBundleForWebServiceOperations` with its default value of `true` sets the application behavior system-wide. It is possible to reverse this behavior (to make it unsafe) on a method-by-method basis by adding the `@WebServiceIgnoreBeanVersionConflicts` annotation to a method. This annotation only works immediately before a method declaration, not a class declaration, and only on classes with the `@WebService` annotation. There is a known issue that Gosu appears to permit `@WebServiceIgnoreBeanVersionConflicts` on the class level without a compile error. However, it has no effect at the class level.

IMPORTANT It is possible that you previously implemented a workaround for this problem by using the `setIgnoreVersionConflicts` method on the `CommitOptions` at the beginning of SOAP implementation methods. If so, then you must update your client-side logic for detecting and retrying in the case of a concurrent change exception to handle the new exception.

Upgrade Issues

This section describes changes to the ClaimCenter base configuration that may cause upgrade issues:

- Display Key Changes
- Document Assistant ActiveX
- Subrogation cookbook implementation can cause upgrade issues (CLM-11150)

Display Key Changes

If you have implemented the localized modules, then you must modify the display keys in your application. Refer to *this file* for details on the changes.

Document Assistant ActiveX

Added a second version of the Guidewire Document Assistant ActiveX control that Guidewire provides to manage many of the functions related to document manipulation. The default version uses a non-configurable whitelist of file types that the Document Assistant control will open. It uses the application associated by the operating system with that file type. If the file type is not on the whitelist, Guidewire Document Assistant will not open the file. The new blacklist version provides a list of known potentially dangerous file types that Guidewire Document Assistant prevents from opening under any circumstances. The blacklist version also includes a configurable whitelist of file types to open. The Document Assistant will open any file type on the whitelist, unless it is also listed on the blacklist. If a file type is on neither list, the Document Assistant prompts the user to save the file.

You can select which version of the Document Assistant to use. If you select the blacklist version, you can configure the whitelist of allowed file types. For information on the default allowed and disallowed file types and how to configure either version of the Guidewire Document Assistant ActiveX control, see the System Administration Guide, topic *Configuring Guidewire Document Assistant*.

Subrogation cookbook implementation can cause upgrade issues (CLM-11150)

Issue: ClaimCenter 4.0 customers that implemented the *Subrogation Cookbook*, article ID # 286 on the *Guidewire Customer Service Center Portal*, may run into difficulties when upgrading unless steps are taken prior to upgrade. The cookbook provides to ClaimCenter 4.0 customers similar subrogation functionality to what was released in ClaimCenter 5.0. If you implemented this cookbook without first modifying the entity names listed in the cookbook, then you will not be able to upgrade to ClaimCenter 5.0 or 6.0 without first resolving a naming conflict.

One way to determine if this issue applies is if any of the following entities are found in the `extensions.xml` file: `SubroAdverseParty`, `StatuteLimitationsLine` or `SubroPaymentSchedule`. If you find those entities but they have a slightly different name, such as `subroadverseparty_ext`, then no conflict should occur.

Workaround: If one or more of those entities are found in the `extensions.xml` file, then you have two options.

- Rename the entities so that no conflict exists. For example, change `SubroAdverseParty` to `subroadverseparty_ext`. Migrate the data and update all related `.pcf` files and Gosu code, *or*
- Migrate the data from the current entities to the default configuration entities.

For detailed information on these options, refer to article ID #643 on the Guidewire Customer Service Center Portal.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Rules Changes
- Improvements to Upgrade Diff Report
- Improvements and General Issues
- Known Issues and Limitations

Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

ClaimCenter release 6.0.3 to 6.0.4

- To view a report of the changes in the base PCF files in the `modules/cc` directory, [click here](#).
- There were no changes to the base PCF files in the `modules/pl` directory.

Rules Changes

ClaimCenter release 6.0.3 to 6.0.4

- There were no changes to the base rules files.

Improvements to Upgrade Diff Report

With previous releases, Guidewire provided a static report in the release notes detailing certain differences between the current release and the prior minor release. This report described changes in display keys, entities, typelists, and the Gosu API. Guidewire has improved this report to dynamically conform to each customer's particular upgrade path. In addition, the report is regularly updated as the tools for generating it are enhanced to provide more information.

Because the new report is tailored to your particular upgrade requirements, it is no longer included with the general release notes. To obtain your custom Upgrade Diff Report, contact your Guidewire representative. In the future, this report will be available on the Guidewire Resource Center web portal, allowing you to view the latest version right when you need it.

Improvements and General Issues

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

ID	Description
Claim	
CLM-12976	Fixed an issue in which the Japanese date format in Calendar was incorrect.
CLM-15821	Previously, changes to the ACL caused updates to the Claim entity, resulting in unexpected ConcurrentDataChangeException exceptions. Guidewire has removed the Claim ownership of the Claim.Access array so that changes to the array no longer trigger Claim updates.
Command-line Tools	
CLM-15486	Fixed an issue that caused a log4j error if trying to check the status of a batch process by its ID. The error message stated: log4j:WARN No appenders could be found for logger(org.apache.axis.i18n.ProjectResourceBundle).
Contacts	
CLM-15288	Modified ClaimContactRole to find existing contacts as appropriate. Previously, method addRole could potentially create a duplicate contact. Now, before saving, ClaimContactRole attempts to remove any duplicates, substituting the existing contact in place of the new duplicate.
CLM-15520	Improved performance related to viewing a contact, in the ClaimCenter Address Book tab or through the Claim Summary page, that has a large number of links to other contacts. Previously, if a user opened the Contact Details page, ClaimCenter issued queries to count all the Activities, Claims, Exposures, and Matters associated with the contact. These queries could be very expensive for a highly linked contact. The performance changes involve the following: <ol style="list-style-type: none"> 1. Changing the pages so that ClaimCenter only executes the queries if the user specifically clicks on the UI card associated with that query. For example ClaimCenter does not query on the Exposure table unless a user explicitly clicks on the Exposures card in the UI. 2. Adding a warning for highly linked contacts. If a contact has more than a certain number of links (configurable), then ClaimCenter issues a warning if the user clicks on a card that can result in an expensive query. The user must click another button to proceed. New configuration parameter, HighlyLinkedContactThreshold, sets the threshold value for the number of links to a contact that generates the warning.
Core	
PL-11508	Corrected a regression that hid a number of methods previously available on Java external entity interfaces. Guidewire has re-exposed these methods.
PL-11585	Corrected an issue in which the application did not recognize a modified version of web.xml, but instead utilized the original version.
PL-11931	Updated the description of properties RateScale and NormalizedRate on the ExchangeRate entity to consider them deprecated. Use the Rate property instead to set exchange rates.
PL-12255	Corrected an issue that could potentially cause a thread deadlock in the GeocodeUtils.getCurrentGeocodePlugin method.
PL-12464	Fixed an issue that caused an 'Attempt to access bean of type "GroupUser" with a null bundle' error when retrying the assignGroupDynamically and the assignUserDynamically assignment methods after a Validation warning.
PL-12989	Fixed an issue that caused the application to not be able to load root group extension definition during initial upgrade of an empty database or from the import_tools command.
PL-13733	Fixed an issue that cause a CDCE exception (ConcurrentDataChangeException) due to stale cache data.
PL-14091	Fixed an issue in which a Claim search using the "Any Party Involved" option did not return the correct response if the value entered for Organization Name could match a Place name.
PL-14130	Corrected an issue that caused policies with a large number of data entities (such common commercial policies) to experience significant performance issues. Policies with root entities that had a large number of child entities (a location that contained a large number of buildings, for example) were especially vulnerable to this type of performance issue.

ID	Description
PL-14322	Fixed an error with French locale using comma as a decimal separator. The CurrencyValueWidget improperly reformatted numbers from French locale to U.S. locale when running verifications.
Database	
PL-12629	0 is now a valid value for the attribute <code>samplingpercentage</code> in <code><databasestatistics></code> element (used with the <code><database></code> element in <code>config.xml</code>). It translates to a statistics estimation percentage of <code>DBMS_STATS.AUTO_SAMPLE_SIZE</code> in Oracle and 100% or <code>FULLSCAN</code> in Microsoft SQL Server.
PL-14389	Corrected an issue with SQL Server that occurred during recreation of primary keys with backing clustered indexes during an upgrade. The process did not account for self-referencing foreign keys as it dropped foreign keys prior to recreating the primary keys.
PL-14446	Corrected an issue that occurred if moving to clustered indexes on SQL Server during an upgrade and using file groups, the application moved the data from the original file group. The application now leaves the data in the original file group when moving to clustered indexes during an upgrade.
PL-14483	On application startup, ClaimCenter checks the database version against the required major version and minimum minor version. ClaimCenter generates an error if the major version does not match or if the minor version is less than the minimum minor version supported. This is a change from previous (pre-7.0.0) releases as later minor release versions would also generate an error. See the “ <i>Database Patch Strategy 2011</i> ” whiteboard or the Product Support Matrix available on the Guidewire support Resource Center for more information.
Financials	
CLM-13289	<p>Simplified the calculation of <code>CheckRpt.GrossAmount</code> field to get the value directly from its <code>Check</code>. This prevents rounding errors that ClaimCenter may have exhibited previously.</p> <p>Note It is possible that the previous calculation of <code>CheckRpt.GrossAmount</code> masked a negative <code>GrossAmount</code> for the <code>Check</code> and instead set the value to zero. Thus, ClaimCenter would show a zero <code>GrossAmount</code> for the <code>Check</code> in the <code>ChecksLV</code>, and show the <code>GrossAmount</code> on the <code>Check Details</code> screen as negative. Now, ClaimCenter shows both fields as negative, which is correct.</p>
CLM-13729	<p>Added consistency checks so that a bulk invoice payee <code>Contact</code> row must enforce the following:</p> <ol style="list-style-type: none"> 1. A bulk invoice payee <code>Contact</code> row must be linked to <code>AB</code> and hence have an address book <code>UID</code>. 2. A bulk invoice payee <code>Contact</code> row belongs solely to the bulk invoice, and is not linked to it by any entity in the claim graph, or else archiving such a claim could fail.
CLM-15714	<p>Added configuration parameter <code>AllowBulkInvoiceItemsToHaveNegativeAmounts</code>. If set to <code>true</code>, then ClaimCenter permits <code>BulkInvoiceItem</code> objects to have negative values.</p> <p>IMPORTANT If you set this value to <code>true</code>, then Guidewire strongly recommends that you configure the application PCF pages and business rules to enforce valid values as there is an increased risk of invalid bulk invoices.</p>
CLM-15418	<p>Fixed an issue in which one or more checks in a recurrence are in <code>Submitted</code> status while the remaining checks are in <code>AwaitingSubmission</code> status and the user edited the recurring check.</p> <p>Guidewire has clarified and corrected the behavior that resulted from this situation:</p> <ul style="list-style-type: none"> • The Edit Check Wizard now modifies the first editable check in a recurrence, instead of attempting to modify a <code>Submitted</code> check, which resulted in an exception. • The system displays an error message if you attempt to reject a recurring check that contains <code>Submitted</code> checks. In this case, edit the recurring check and set the number of checks in the recurrence to match the number of <code>Submitted</code> checks. This action deletes the undesired checks from the recurrence. You can also delete non-<code>Submitted</code> checks individually.
CLM-15463	Removed dropping and recreating indices during staging table loading. Dropping/recreating indices performed an implicit commit on Oracle, which committed the staging table load early. The fix preserves the ability to rollback transactions if needed. For example, ClaimCenter now typically inserts <code>FinancialsRpt</code> information without dropping and recreating indices.
CLM-15585	Added new configuration parameter, <code>EnablePreSetupRulesInCheckCreator</code> , that you can use to invoke Pre-setup rules on the check set by the <code>CheckCreator</code> as it creates a check. The default is <code>false</code> .
CLM-15476	<p>Added an enforcement that the Total Payments financials calculation cannot go negative while editing an existing check in the <code>Check</code> wizard. This fix ensures that the <code>Claim</code> and <code>Exposure</code> financials are still valid as part of any update.</p> <p>Previously, ClaimCenter did not display the error message at the appropriate time, during check editing. Now, ClaimCenter does display the error message correctly.</p>
Gosu	

ID	Description
PL-10812	Corrected an issue in which a finder in a code annotation caused Studio to throw an error, and fail to perform syntax error highlighting and code completion. This occurred even if Studio was connected to the application server (meaning the database).
Integration	
PL-14984	See "Change in Web Service Behavior for Committing Data to the Database (CLM-15982)" on page 552 for details.
Miscellaneous	
CLM-15201	Modified the ClaimCenter Vehicle Details screen to remove hard-coded List View labels for the Driver and Passenger fields.
CLM-15612	Updated <code>CountryAddressFields.gs</code> to be null-safe for the current country. As a workaround for previous versions, you can add the following to <code>CountryAddressFields.forCountry(currentCountry : Country)</code> : <pre> if (currentCountry == null) { return new CountryAddressFields(currentCountry) } </pre>
CLM-15653	Fixed an issue in which ClaimCenter did not populate the description field properly as it created a document from a template and the description existed in the template descriptor.
CLM-15793	Fixed an issue that caused a <code>NestedTransactionException</code> error if <code>ISO.ExpectReplies</code> was set to <code>false</code> while using ClaimCenter 5.0 <code>ISOMessageTransport</code> implementation for the ISO plugin. The issue was that a new transaction bundle was used to acknowledge the ISO response and update the <code>ISOKnown</code> field when one existed already.
Performance	
CLM-15723	Replaced <code>OR</code> with <code>UNION</code> in a poor-performing query to improve <code>ClaimException</code> performance.
Policy	
CLM-15465	Corrected an issue in the Lines of Business editor that caused an <code>EvaluationException</code> exception while creating a new <code>LossType</code> .
Reporting	
CLM-13023	Modified the formatting of the Claims Processing Effectiveness report to disallow decimal values in numeric columns.
PL-14418	Corrected an issue that caused a report to freeze if that report contained freehand tables along with large data sets. This issue was due to a bug in the InetSoft v9 reporting software.
CLM-14961	Fixed an issue in which drilling down to the Claim Catastrophe Detail report from the Catastrophe Financials by Claim Owner report displayed an incorrect message stating there were no records when indeed, records existed.
CLM-14962	Fixed an issue in which drilling down to the Catastrophe Financials by Claim Owner report from the Catastrophe Financials report did not correctly include claim-level transactions.
CLM-15300	Modified the Monthly Recoveries report to limit the transaction types viewable in the report to Recovery only.
CLM-15413	Modified the formatting of the First Payment Productivity report to disallow decimal values in numeric columns.
CLM-15420	Corrected an issue that caused the following warning message to show below the footer of the last page of the Loss Run report. "Element "Table1" is printed with max number of rows: 1"
Studio	
PL-10701	Improved performance with regards to the amount of time that it took to create a new class in Studio.
PL-12877	Fixed an issue that caused a <code>Null Pointer Exception</code> and Studio to stop responding. This occurred if you attempted to create a new rule category with the same name, but different case, of an existing rule category.
PL-13486	In Studio, as you switch between tabs, close tabs, press ALT-F1 to find in the project tree, and perform other related actions, Studio performs a search to find the node in the project tree that corresponds to the active editor. This fix improves performance in Studio as you switch between tabs, close tabs, and perform similar actions.

ID	Description
PL-13623	Fixed an issue where the PCF editor object was not garbage-collected after the PCF editor window was closed. This could cause sluggish Studio behavior or possibly even out-of-memory issues in long Studio sessions in which you created or modified a large number of PCF files.
PL-13626	Improved performance with respect to creating new classes, new PCF files, a new rule set, and new rules in a new rule set in Studio.
PL-14214	Reinstated the requirement that a Studio user must have the Administer Rules permission (code, ruleadmin) to access a running production server.

Known Issues and Limitations

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

- ClaimCenter Known Issues
- Platform/Studio Known Issues

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

Exception in the Bulk Invoice screen after restoring and reopening an archived claim (CLM-174)

Issue: If you archive a claim, restore it within 60 minutes of archiving it, and then immediately attempt to view a bulk invoice involving that claim, the system may show an exception message. This is because it takes time for the global cache to be updated after the claim is archived. However, this does not cause any data loss or corruption.

Workaround: You must wait 60 minutes after archiving a claim before restoring it. The length of time to wait is determined by the `GlobalCacheStaleTimeMinutes` configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt to view the bulk invoice for 60 minutes. Guidewire may not fix this issue, since restoring archived claims so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

Link issues between a new contact from the New Check wizard and ContactCenter (CLM-12172)

Issue: If you try to link a contact newly created from the New Check wizard to ContactCenter, the contact details can be overridden if the contact already exists in ContactCenter. This occurs if there is a match between the two integrated systems. Rather than creating a new contact, the existing matching contact information in ContactCenter overwrites the contact's information in ClaimCenter.

Deleting an user who owned an archive claim and running the archive worker item throws a `DBDuplicateKeyException` (CLM-12868)

Issue: If a user is deleted after a claim has been archived and restored, then the archive worker item throws a `DBDuplicateKeyException` exception.

Workaround: Set the user who owns an archive claim to inactive instead of deleting them.

Tax ID fields are not masked in claim snapshots that were created in earlier versions of ClaimCenter (CLM-12879)

Issue: Tax ID fields in the claim snapshot are not masked in the user interface. This happens because the claim snapshot is using an earlier version of the `ClaimSnapshotContactxxx.pcf` file which does not have the field encryption on the claim snapshot. (xxx represents various versions of the pcf file.)

Workaround: You must manually add a mask to the encrypted fields in earlier versions (such as 5.0.0) on the `ClaimSnapshotContactxxx.pcf` file.

Workaround: Create the contact first in ClaimCenter (not in the New Check wizard), then if you want, link it to ContactCenter. Finally, use the New Check wizard to create a check for that contact.

Unable to generate a Claim Metrics report through the InetSoft Report Portal (CLM-13141)

Issue: The InetSoft onLoad script is designed to be run from within ClaimCenter and it should pass the report the locale. If the report is run outside of ClaimCenter, the locale needs to be set within InetSoft.

Workaround: Run reports through ClaimCenter.

Restoring a claim fails if there is no assignment and the default owner is not in a group (CLM-13205)

Issue: A null pointer exception is thrown in the user interface if you removed the default owner from the root group, and you attempted to assign the claim.

Workaround: Put the default owner in a group, preferably the root group (default).

Integrity and consistency checks on multicurrency amounts are not complete (CLM-13243)

Issue: The integrity and consistency checks on multicurrency amounts are not yet as strict as the enforcement that exists when new transactions are created in ClaimCenter.

This is specifically for the entities: `TransactionLineItem`, `Deduction`, and `CheckPortion`. Any two amounts on an entity should be equal when they are in the same currency. For example, for an instance of `TransactionLineItem`, `TransactionAmount` must equal `ClaimAmount` if `Transaction.Currency==Transaction.Claim.Currency`. It is similar for `TransactionAmount-ReportingAmount` and `ClaimAmount-ReportingAmount`. For `CheckPortion`, the enforcement is only performed if `FixedTransactionAmount`, `FixedClaimAmount`, and `FixedReportingAmount` are not null.

Workaround: For multicurrency implementations, manually ensure during the population of staging tables that the amounts on the entities `TransactionLineItem`, `Deduction`, and `CheckPortion` are consistent, given their currencies. For single-currency implementations, all amounts should be the same, avoiding the issue.

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.

Status: Guidewire is aware of this issue.

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 goes down the left side instead of across.

Status: Guidewire is aware of this issue.

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.

Localized document pattern specified in the activity pattern is not passed to the activity (CLM-13670)

Issue: When ClaimCenter is localized to the Japanese language, and you create an activity using the *Gosu Sample Email Sent Record* activity pattern, the system does not select the correct language.

Workaround: Use the document search feature to find the applicable localized document template.

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().

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, then continue to use the deprecated methods.

Must regenerate database statistics after large batch processing of addresses (CLM-13963)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the ab_abaddress table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

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)

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.

Duplicate descriptions for image/jpg mime types in the configuration file (CLM-14188)

Issue: In the file `modules\pl\config\locale\en_US\display.properties`, there are two entries under MIME types for jpeg file types that are defined with the same description, **JPEG Image**:

- `pjpeg`
- `jpeg`

These definitions cause a problem because the descriptions of both mime types are identical. If you have the **New Document** screen open to add a new image document and you click the **Document Type** drop-down list, you see two entries for **JPEG Image**. If you choose the wrong image type and add the new document, an error occurs.

Workaround: Open Guidewire Studio and do the following:

1. Navigate in the left pane to **Configuration** → **Display Keys** → **Mimetype** → **Configuration** → `image_pjpeg`.
2. Change the **Locale: English (US)** text to **Progressive JPEG Image**.
3. Save your changes and restart ClaimCenter.
4. Now in the ClaimCenter **New Document** screen, when you click the **Document Type** drop-down list, you see two choices for these file types, **JPEG Image** and **Progressive JPEG Image**.

No warning that you will overwrite a Service Provider Review or Question Set (CLM-14365)

Issue: If you do not retire the review or question set and import the new review or question set, then there is no warning or mechanism to prevent you from overwriting the existing review or question set.

Workaround: You should create the new review or question set in a new `.csv` file. Also, ensure that all public IDs of all entities of the new review or question set are unique and different from the existing review or question set. Consider that any entities such as `ReviewType`, `ReviewCategoryQuestionSet`, `QuestionSet`, `Question`, `QesitonChoice`, and `QuestionFilter` could be overwritten.

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.

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.

Drill down to Claim Catastrophe Detail report displays “No Records Found” when there are claims (CLM-14961)

Issue: If you drill down from **Claim Group** to the **Catastrophe Financials by Claim Owner** report and then clicking on a **Claim Owner** to access the **Claim Catastrophe Detail** report, ClaimCenter displays an incorrect message stating there are no records when indeed, records exist.

Status: Guidewire is aware of this issue.

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.

Snapshot encryption upgrade batch process fails (CLM-15527)

Issue: The snapshot upgrade work queue (com.guidewire.cc.system.crypt.SnapshotUpgradeWorkQueue in the work-queue.xml file) throws a ClassCastException when it is attempting to retrieve the entity type of an object snapshot. This issue can only occur if you upgrade your encryption plug-in or change your encrypted fields, and only prevents the upgrading of existing snapshots to the new encryption configuration.

Workaround: None. It does not affect the ability to view those snapshots in ClaimCenter.

UnsupportedOperationException: MetaIntrinsicType does not support array yet (CLM-15718)

Issue: ClaimCenter 3.x snapshot PCF files generate a MetaIntrinsicType does not support array yet exception if the snapshot PCF contains the following cell:

```
Cell id="Lienholders" value="util.Snapshot.renderValue(PolicyVehicle.Vehicle.Lienholders)"
label="displaykey.Web.ClaimSnapshotPolicyVehicles300LV.Lienholders"/>
```

You see this problem if you navigate to the **FNOL Snapshot** → **Policy** → **Vehicle** screen and select a ClaimCenter 3.x snapshot claim that contains this particular cell.

Workaround: Navigate to the following Studio location and comment out the cell in any PCF file that contains it.

Page Configuration (PCF) → claim → snapshot → 300

Entering a money amount such as “.23” displays a validation error (CLM-16095)

Issue: In the base configuration, Guidewire defines the field validator regular expression for Money columns as the following:

```
-?[0-9]{1,16}(\.?.?.....
```

However, due to a fix that Guidewire added for PL-14322, passing a value such as .23 generates a validation error. The issue is that the regular expression requires at least one digit before the decimal separator.

Workaround: Modify the field validator regular expression for the Money column so that {1,16} becomes {0,16}. For example:

```
-?[0-9]{0,16}(\.?.?.....
```

Platform/Studio Known Issues

ClaimCenter does not display empty report folders (PL-1281)

Issue: ClaimCenter does not display a report folder in the **Administration** → **Report Admin** page if that report folder is empty.

Workaround: Within InetSoft Enterprise Manager, insert a replet (report template) into the empty report folder. You can then mark this replet as not visible (using the InetSoft Enterprise Manager), which causes the replet to not show in ClaimCenter. However, ClaimCenter does display the parent folder.

SampleAcrobat document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in SampleAcrobat.pdf.descriptor and SampleAcrobat.pdf files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Guidewire Studio PCF Editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Status: Guidewire is aware of this issue.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because ClaimCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

ClaimCenter cannot load zone data with missing values (PL-9123)

Issue: Zone data is used for associating postal codes with cities and regions. You cannot load zone data that has empty fields.

Workaround: Guidewire is aware of this issue. As a temporary workaround, you can enter a character into that field, for example, a hyphen (-).

H2 development database creates LONGTEXT fields as VARCHAR (65000), instead of as CLOB as in Oracle and SQL Server (PL-9314)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database. Note that the H2 database is *only* used for testing and should never be used in production.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

License state drop-down menu in the user interface produces unexpected results (PL-9690)

Issue: If you use the `AddressAutocompleteUtil.getStates()` method to populate license states in the `FNOLVehicleIncidentPopup.pcf` file, then you might see unexpected results, such as countries or retired type-codes. This occurs when you select the license state drop down menu in the user interface. What is happening is that `AddressAutocompleteUtil.getStates` is passing true to the `getTypeKeys` method when it should be false.

Workaround: Use the following:

```
function getStates_Ext(country : typekey.Country) : typekey.State[]{
    var stateList = new java.util.ArrayList()
    var allStates = typekey.State.getTypeKeys( false ) //gets all states except the retired ones
    if(country==null){
        country = &quot;US&quot; //default to US
    }
    for(eachState in allStates){
        if(eachState.hasCategory( country )){
            stateList.add(eachState)
        }
    }
    return stateList.toArray() as typekey.State[]
}
```

Include this function in your PCF file. Instead of calling the `AddressAutocompleteUtil` method in the `valueRange`, call the following method instead:

```
valueRange = getStates_Ext(VehicleIncident.LocationAddress.Country)
```

Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled (PL-9813)

Issue: If you have the Windows Server 2003 component called IE Enhanced Security Feature installed, tabs may be unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server.

Country specific field validation does not run if the fields are not modified when a country is changed (PL-9828)

Issue: There may be an error if you change a contact field in the user interface that is validated with country specific field validators. Those specific validators do not get revalidated if the country changes and the field is not modified. This issue only occurs if you use country specific field validators.

For example, suppose that a carrier operates in two countries, the United States and Canada and wants to implement different field validation for certain data types such as Tax ID. If an adjuster changes the country on an existing contact record from the United States to Canada, without modifying the Tax ID to reflect the new format, the system allows the US formatted Tax ID to be saved. If the adjuster *edits* the Tax ID, then the system revalidates the format.

Workaround: Use the Reject method--in one of its several forms--in a validation business rule to validate critical fields. ClaimCenter runs the validation rules any time that it commits a business object that implements the Validatable delegate to the database. For this case, create a validation rule that checks whether the country field has changed without other specific fields changing, and then generate an appropriate error message.

Not possible to hot-deploy on all supported application servers (PL-9955)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

First time you click on the arrow of the typekey input, the drop down menu will not open (PL-10134)

Issue: The drop down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting InputHelpTextOnFocus to false in the config.xml file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

InetSoft Exception *Failed Login Exception: Bad user name anonymous* (PL-10148)

Issue: InetSoft generates a Failed Login Exception: Bad user name anonymous exception on server start-up. This occurs if you set the InetSoft logging level to *fine*.

Workaround: This error does not cause any issues. You can safely ignore the exception.

Report still visible after permissions disabled for parent folder (PL-10353)

Issue: An InetSoft report that is included in a permission set is still visible in the report tree even after its parent has been removed from the permission set.

Workaround: You must remove individual reports from permission sets instead of maintaining the reports at the report folder level.

Studio Rules do not use correct capitalization for root object's name (PL-10740)

Issue: Ruleset root objects are not named with first letter lower-cased.

Workaround: Rules engine will issue warnings when correct case for objects is not being used.

ToolBarButton widget does not render correctly (PL-10742)

Issue: ClaimCenter incorrectly renders a *ToolBarButton* widget if the associated *AddMenuItemIterator* is empty.

Workaround: Modify your configuration to disable the menu button if the button contains no menu item.

Must name row iterator element with capital letter (PL-10795)

Issue: The Document search results page (DocumentLV.pcf) throws a `NullPointerException` if the row iterator element name is *document* (with a lower case d).

Workaround: You must use *Document* (with a capital D) instead.

Studio does not properly report verification error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Navigating between steps while in a popup in a wizard may lose data (PL-10920)

Issue: If you are in a wizard and enter a popup page, then type in data but do not save, then navigate directly to a different step in the wizard by clicking that step's link on the left side, the data that was entered in the popup page and not saved may be lost. For example, this can occur if you are in the FNOL wizard on the **Exposures** step, and you add a new exposure, which takes you into a popup page. If you enter data on that **New Exposure** page, then navigate directly to a different wizard step, such as **Loss Details**, the data you entered into the **New Exposure** page will be lost.

Workaround: Ensure when entering data in a wizard that you always click **Update** (or **OK** or **Save**) to return to the parent wizard step, before navigating to a different step of the wizard.

Values in MIME-type drop-down menus are not localized (PL-10921)

Issue: ClaimCenter does not localize the values in the **MIME type** drop-down list properly.

Workaround: If you want to localize MIME types, you can do so by modifying the definition in `config.xml`. Currently, ClaimCenter does not support MIME type definitions in multiple languages.

Studio—with Subversion (SVN)—incorrectly copies .svn files (PL-10932)

Issue: *This issue affects installations that use Subversion only.* If you modify the base configuration rules in Studio, Studio creates a copy of the rule in the `modules/configuration` folder. Studio then executes Subversion commands to add the files and folders to Subversion. In the process, Studio incorrectly copies the `.svn` folder and its children. This leads to Subversion problems.

Workaround: Manually delete the copied `.svn` folders

Double quotes in *PATH* environment variable causes problems (PL-10981)

Issue: ClaimCenter does not run if there are double quotes in the *PATH* environment variable.

Workaround: Check the *PATH* variable and remove any double quotes.

Large number of classes in *plugins\shared\lib* causes studio to fail to start (PL-11026)

Issue: If you attempt to load multiple JAR files that contain many thousands of classes, Studio fails to start and throws an `OutOfMemory` error.

Workaround: Increase `MaxPermSize` to something over 128m (the default) so that Studio has enough memory to build the type information database.

ANT command *build-war* hangs indefinitely (PL-11045)

Issue: It is possible for the `build-war` ANT command to hang indefinitely. This issue is only reproducible in certain environments.

Workaround: Set the `-noinput` parameter before calling the `build-war` ANT command. For example, instead of `gwcc.bat build-war`, do the following:

```
gwcc.bat -noinput build-war
```

Cache summary statistics rely on server ID being unique (PL-11239)

Issue: If you use the same server ID for multiple servers in a cluster, it can create the following problems:

- Different servers in a cluster deleting or changing each others results
- Deadlocks at the database level

Workaround: By default, the server ID defaults to the host name of the machine running the instance. This works well in many configurations, but causes problems if multiple server instances are running on a single machine. Set the `serverid` system property in the `config.xml` file on the local server host to a unique value in the cluster. If it does not currently exist, then you need to add a `<registry>` element to the `config.xml` file on the local server host.

You can also start the server with the following JVM option: `gw.cc.serverid=????`

See *Defining the Application Server Environment* in the System Administration Guide for details on both of these options.

Geocode plugin calls *isSufficientlyCompleteToGeocode* method twice for each address (PL-11578)

Issue: The Geocode plugin calls the `isSufficientlyCompleteToGeocode` method twice for each address.

Workaround: Guidewire is aware of this issue.

Missing display keys cause server to fail to start (PL-11764)

Issue: Missing display keys defined in the application default locale can cause the server to fail to start.

Workaround: Within Guidewire Studio, run the **Verify Resources** utility before attempting to start the application server. ClaimCenter bases the type system for display keys on those defined in the default application locale. If a display key in the default application locale is missing and a rule references the missing display key, then ClaimCenter cannot compile the rule. Running the resource verification utility can catch these types of missing display keys.

Countries configured in `zone-config.xml` still generate a warning during regen-dictionary even 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.

Hidden reports should not be visible in Report Admin tree (PL-12313)

Issue: If you set a report's visibility to false in InetSoft, you can still see the report in the ClaimCenter **Admin Report** tree. The path is **Administration** tab → **Report Admin**. However, these reports still remain hidden to any report users from both the InetSoft Portal and in ClaimCenter.

Workaround: Give the Rule Admin role (which gives permission to access the **Rule Admin** screen) to only those users for which seeing all reports is not an issue.

Application interface cannot handle starting multiple instances of a batch process (PL-12372)

Issue: The user interface cannot handle multiple instances of a batch process.

Workaround: If multiple instances of a batch process need to be executed, they must be started from the command line. Also, you should ensure that the `BatchProcess.isExclusive()` method returns `false` to allow multiple instances to be run simultaneously.

The .csv file is incorrectly configured in mimetype mapping (PL-12841)

Issue: In ClaimCenter, the .csv file is in a different format than what is specified by the file extension.

Workaround: Update the MIME type in your `config.xml` file.

Debugger break point for the rule engine does not work if first the line is a comment or a blank line (PL-12912)

Issue: The debugger break point for the rule engine does not work if the first line is a comment or a blank line.

Workaround: Ensure that the first line is not blank and does not contain a comment. Add comments after the first line.

Type system refresh after PCF page title change does not update corresponding menu label (PL-13057)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

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.

US-Locations.txt file with the US geodata from GreatData has special characters that cause validation problems with United States Postal Service (USPS) data (PL-13384)

Issue: The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can process the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

Use of *StringBuilder* class prevents debugging of web service (PL-13435)

Issue: Suppose that you have a web service in which you have a method with a parameter of type `X` and inside the web service method you reference a Gosu class of type `Y`. This can be an issue if the Gosu class `Y` has a `toString` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you attempt to set a breakpoint in the web service method and try to step through the method, Studio generates errors.

Workaround: Studio does not generate errors if you call the web service in normal run mode rather than in debug mode.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

Sending email with file attachment with Unicode filename is not correctly handed over to the mail server (PL-13582)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

The GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

JavaToolkit.gs has incorrectly hard coded memory which results in failed *regen-java-api* ANT task (PL-13663)

Issue: The JavaToolkit.gs has hard coded memory which can result in failed *regen-java-api* ANT tasks.

Workaround: Increase the size of the max heap setting on line 161 of JavaToolkit.gs in the Ant module. The shipped value is 512.

Cannot make a field from a delegate a localized column (PL-13761)

Issue: You cannot make a field from a delegate a localized column.

Workaround: Move the column to be localized off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define an enhancement property on the delegate, that *delegates* to the appropriate column, depending on the implementing entity.

Copy and paste does not work from phone fields (PL-13792)

Issue: The copy and paste does not work for phone fields on a contact.

Workaround: Save the contact with the newly entered phone information before attempting to either copy or paste the value into another field.

Cannot deploy custom *web.xml* file (PL-14482)

Issue: ClaimCenter does not deploy a custom *web.xml* file properly.

Workaround: Before attempting to deploy the application, do the following:

1. Open the following file for editing:

```
ClaimCenter\modules\ant\public\gw\ant\deploy.gs
```

2. Search for the following lines of code:

```
var deployDir = module.Dir.file("deploy")
if (deployDir.exists()) {
  new Copy() { :FileSet = deployDir.fileSet(), :ToDir = destDir }
  .execute()
}
```

3. Make the following change:

```
var deployDir = module.Dir.file("deploy")
if (deployDir.exists()) {
  new Copy() { :FileSet = deployDir.fileSet(), :ToDir = destDir, :Overwrite = true }
  .execute()
}
```

Notice that you need to add `:Overwrite = true` to the Copy method.

Cannot print second-level list view (PL-14640)

Issue: Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all of the items.

Workaround: Remove the lower level list view out of the hierarchical structure and include it at the top level.

Upgrade from ClaimCenter 4.x to ClaimCenter 6.x upgrades Regions but not associated Region Zones (PL-14761)

Issue: The upgrade from ClaimCenter 4.x to ClaimCenter 6. upgrades Regions. However the upgrade process does not bring the associated Region Zones along with the Regions.

Workaround: Manually add the zones. You can also do a file import of the zones using `import_tools`.

Superuser role unable to edit or delete calendar holidays (PL-14942)

Issue: It is not possible to edit or delete a calendar holiday as superuser.

Workaround: Log into ClaimCenter using `su` to be able to edit the holiday schedule.

RowSet and RowIterator configured in the same ListView do not work correctly. (PL-14946)

Issue: Configuring a RowSet and RowIterator in the same ListView does not provide the right behavior for editability.

Workaround: Separate the RowSet and RowIterator that exist in the same ListView into separate components or into multiple ListView widgets.

Masked input fields do not correctly handle ENTER keypress (PL-14955)

Issue: Input fields that have a field validator mask do not correctly handle the ENTER keypress. After you press ENTER, the field no longer displays the input mask and it is not possible to enter any further input.

Workaround: Cancel the page and re-enter the information.

ClaimCenter does not recognize modified Calendar.js file (PL-14997)

Issue: ClaimCenter does not correctly recognize changes made to `Calendar.js` through Studio. You access this file in the following Studio location:

Resources → Web Resources → resources → javascript → global

Workaround: Instead, modify `global.js` directly. You access this file in the following installation file location:

ClaimCenter/webapps/cc/resources/javascript

WARNING This is one of the very few occasions in which you modify an application file outside of the application configuration folder. Ensure that you modify `global.js` only. Otherwise, you can invalidate your installation.

Running regen-java-api gives “plugin not found” error (PL-15742)

Issue: When you run `gwcc regen-soap-api` or `gwcc regen-java-api`, you get a “gw-cc-plugin not found” error.

Workaround: Manually create the `doc/gw-cc-plugin` directory, and then try again.

chapter 38

Guidewire ClaimCenter 6.0.5 Release Notes

Release 6.0.5.39

Copyright © 2001-2013 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Live, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

This material is Guidewire proprietary and confidential. The contents of this material, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

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

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

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
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 6.0.5.39.

For Standard Reporting, this release supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter
- Prior ClaimCenter release notes for any versions that have been skipped

The following issues pertain to installing this release.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-72C-ERX-0000BF00100001F-F2B025280327

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Issues and Major Changes

This section contains issues or changes that may affect your installation:

- Changes to Archive Functionality

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.

With the release of ClaimCenter 6.0.5, 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 a XML document.

Note: For more information, see “Archiving Claims” in the *ClaimCenter Configuration Guide*.

(CLM-16635) 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 configuration parameter `DaysRetrievedBeforeArchive`
- Removed configuration parameter `ArchiveServer`.

(PL-15380) Guidewire has removed the following columns from all base configuration entities on which they existed:

- `ArchiveID`
- `Overlap`
- `PartitionID`
- `ArchiveAdmin`
- `ArchiveTypeKey`

(PL-15004) As part of the upgrade to ClaimCenter 6.0.5, Guidewire removes the `AdminTable` delegate from customer extensions.

(PL-14565) Guidewire has set `extendable="false"` and `final="true"` in the following entities in the base configuration:

- `ArchiveWorkItem`
- `ArchiveFailureDetail`

Guidewire has deleted the following entities from the base configuration:

- `ArchiveTypeKey`
- `ArchiveTransitionRecord`
- `ArchiveAdminKey`

(PL-15064) Guidewire has added new 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 (Server Tools) **Domain Graph Info** page (Warning tab).

Note: For a more complete list of archive-related changes, see the **Archiving** section under “Improvements and General Issues” on page 576.

Support

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

- At the Guidewire Resource Center – <http://guidewire.custhelp.com>
- By email – support@guidewire.com
- By phone – +1-650-356-4955

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Rules Changes

- Improvements to Upgrade Diff Report
- Improvements and General Issues
- Known Issues and Limitations

Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

ClaimCenter release 6.0.4 to 6.0.5

- 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/pl` directory, [click here](#).

Rules Changes

All links below require the `readme_files` directory on your local disk.

ClaimCenter release 6.0.4 to 6.0.5

- To view a report of the changes in the base rules in the `modules/cc` directory, [click here](#).

Improvements to Upgrade Diff Report

With previous releases, Guidewire provided a static report in the release notes detailing certain differences between the current release and the prior minor release. This report described changes in display keys, entities, typelists, and the Gosu API. Guidewire has improved this report to dynamically conform to each customer's particular upgrade path. In addition, the report is regularly updated as the tools for generating it are enhanced to provide more information.

Because the new report is tailored to your particular upgrade requirements, it is no longer included with the general release notes. To obtain your custom Upgrade Diff Report, contact your Guidewire representative. In the future, this report will be available on the Guidewire Resource Center web portal, allowing you to view the latest version right when you need it.

Improvements and General Issues

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

ID	Description
Administration	
CLM-16057	Modified the <code>TeamGroupMatters</code> and <code>TeamUserMatters</code> pages so that they, by default, display all open matters. Previously, these pages displayed all matters, whether open or closed. There is now a filter to choose whether to view All Open , or New Open (this week) , or Closed Last 90 Days .
Archiving	
CLM-13205	Fixed an issue in which restoring an archived claim fails if there is no assignment and the default owner is not in a group.
CLM-13401	Fixed an issue in which ClaimCenter displayed archiving functionality in the user interface even if archiving was not enabled.
CLM-13528	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.

ID	Description
CLM-15941	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 does the following: <ul style="list-style-type: none"> • It creates a new instance of <code>ClaimInfo</code>. • It sets its <code>ArchiveState</code> to <code>archived</code>. • It sets the non-scriptable fields on the new <code>ClaimInfo</code> entity based on the arguments.
CLM-16010	Added a subject for Notes created by archiving issues.
CLM-16027	Added Archived and Restored filter options to the Claim History drop-down list.
CLM-16069	Added a class, <code>PurgeClaimInfoMethodsImpl</code> , that you can edit to return the IDs of objects that are outside the claim graph—especially those linked to <code>ClaimInfo</code> —that you need to delete when you purge the claim.
CLM-16105	Added a new sample Archive rule (ARC06000 - Unsynchronized Review Rule) that instructs the archive process to ignore claims with vendor reviews that have not already been synchronized with the Address Book.
CLM-16128	Added an archive rules that prevents ClaimCenter from archiving a claim if it has any transaction that is not in one of the following states: <ul style="list-style-type: none"> • Submitted • Rejected • Voided • Stopped • Recoded • Transferred • Denied
CLM-16143	Added a new sample Archive rule (ARC05000 - IncompleteReviewRule) that instructs the archive process to ignore claims with incomplete reviews.
CLM-16149	Fixed an issue that caused an <code>IllegalArgumentException</code> if you attempted to retrieve an archived claim multiple times.
CLM-16240	Fixed an issue in which a simple claim search searched the archive database only if you set Advanced Search to search the archive database.
CLM-16423	Fixed an issue in which transferring a bulk check to another claim caused the <code>ReserveLineWrapper.ReserveLine</code> foreign key on the old claim to incorrectly refer to the <code>ReserveLine</code> on the new claim.
CLM-16435	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-16556	Fixed an issue in which ClaimCenter threw an exception if an attempt was made to refresh the policy period and that policy period also referenced archived claims.
CLM-16573	Fixed an issue in which ClaimCenter displayed an exception if an invalid claim number was specified for <code>-scheduleforarchive</code> .
CLM-16579	The default ClaimCenter configuration now requires that a user enter a comment in order to restore a claim.
CLM-16621	Simple Search for claims does not search on archived claims if you specify search fields that are not defined on the denormalized claim information.
CLM-16635	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 <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 <i>Configuration Guide</i> .
CLM-16711	Added the ability to purge archived claims using the maintenance command line tools.
PL-13580	As part of the changes to archiving, Guidewire has removed the following columns from all base configuration entities on which they existed: <ul style="list-style-type: none"> • <code>ArchiveID</code> • <code>Overlap</code> • <code>PartitionID</code> • <code>ArchiveAdmin</code> • <code>ArchiveTypeKey</code>

ID	Description
PL-14565	<p>As part of the changes to archiving, Guidewire has set extendable="false" and final="true" in the following entities in the base configuration:</p> <ul style="list-style-type: none"> • ArchiveWorkItem • ArchiveFailureDetail <p>Guidewire has deleted the following entities from the base configuration:</p> <ul style="list-style-type: none"> • ArchiveTypeKey • ArchiveTransitionRecord • ArchiveAdminKey
PL-14943	Changed how ClaimCenter handles archiving. Guidewire now serializes data objects during archiving, making it possible to store the objects as XML documents in the storage venue of your choice. Guidewire no longer stores archived objects in a database. Instead, you now store archived objects as files.
PL-14983	Modified archiving by removing the Admin graph. Archiving now uses the concept of referenced objects. A reference object is a data object that multiple instances of a domain graph object all share. For example, multiple Claim objects can share the same User data.
PL-15004	As part of the upgrade to ClaimCenter 6.0.5, Guidewire removes the AdminTable delegate from customer extensions.
PL-15060	Added a new section Issues from past 24 hours to the (Server Tools) Archive Info page. This content includes information on those objects that the archive process skipped because of rules or because of a failure of the archive process.
PL-15061	Added an Archive Source section to the (Server Tools) Archive Info page. This includes information on the last time that the page was refreshed and the status of the store, for example.
PL-15062	Added a new Archive Summary by Datamodel Version section to the (Server Tools) Archive Info page. This screen displays the count of RootInfo objects archived at various data model versions.
PL-15064	<p>Added two new configuration parameters for use with archiving:</p> <ul style="list-style-type: none"> • 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. <p>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 (Server Tools) Domain Graph Info page (Warning tab).</p> <p>IMPORTANT Do not use whitespace characters in the value string.</p>
PL-15260	Improved the error messages that results if you attempt to archive a previously excluded claim for the second time.
PL-15720	Changed the name of the ExtractReady column on the Extractable delegate to ArchivePartition.
PL-15737	Modified (Server Tools) Archive Info page so that it now displays the list of archived claims under the Overview section of the screen.
PL-15855	<p>Guidewire now creates a separate log for successfully archived objects. Each log is unique to an archive run. The log contains a list of the claims that were successfully archived.</p> <p>To configure the log, add the following to logging.properties.</p> <pre>##### Archived Claims ##### # # Set up the logging for Archived Claims. log4j.category.Server.Archiving.Success=INFO, ArchivedClaimsLog log4j.appender.ArchivedClaimsLog=org.apache.log4j.DailyRollingFileAppender log4j.appender.ArchivedClaimsLog.File=/tmp/gwlogs/ClaimCenter/logs/archivedClaims.log log4j.appender.ArchivedClaimsLog.DatePattern = .yyyy-MM-dd log4j.appender.ArchivedClaimsLog.layout=org.apache.log4j.PatternLayout log4j.appender.ArchivedClaimsLog.layout.ConversionPattern=%-10.10X{server} %-4.4X{user} %d{ISO8601} %p %m%n</pre>
PL-15871	Modified the behavior of the (Server Tools) Archive Info page so that selecting a claim listed under Excluded Because Of Failure now drills down into a claim summary page, from which you can drill down into the cause of the failure of the claim to archive.

Claim - Summary, Loss Details, Exposures, Incidents, Litigation, Etc

ID	Description
CLM-15934	<p>Added database null and uniqueness constraints on SubrogationSummary.ClaimID to prevent duplicate instances of SubrogationSummary for a single claim. If any one of these constraints is violated, the upgrade fails. Guidewire strongly recommends that customer who upgrade their installations—especially customers upgrading between major application versions—run the following database queries before beginning the upgrade:</p> <pre>select ID from cc_subrogationsummary where ClaimID is null and Retired=0</pre> <pre>select c.ClaimNumber, COUNT(*) as '#(subrogationsummary)' from cc_subrogationsummary a inner join cc_claim c on c.ID = a.ClaimID group by c.ClaimNumber, a.Retired having COUNT(*) > 1</pre> <p>If either of those queries returns any rows, the upgrade will fail, and you need to take additional steps to clean up your data:</p> <ul style="list-style-type: none"> • If the first query returns data (which is unlikely), then you need to delete those SubrogationSummary objects or link them to Claims. • If the second query returns data, then refer to Knowledge Base article 1386, section A for instructions on cleaning it up. You do not need to follow the instructions in section B of the Knowledge Base article, as that section describes applying the index that ClaimCenter creates automatically during the upgrade.
CLM-16632	Fixed an issue in which switching to a currency with a scale attribute of zero could cause an exception on the ClaimCenter (Plan of Action) Evaluations page.
Claim Metrics	
CLM-16624	Corrected a performance issue with the ClaimHealthCalculatorBatch batch process.
CLM-16626	Fixed an issue in which running the 'Recalculate Claim Metrics' Claim Exception rule caused all processed claims to recalculate metrics against the latest version of the metric limits. The required behavior for this claim exception rule is to only create new claim metrics on claims with existing metrics.
ContactCenter	
CTC-288	<p>Added a batch process, Update MatchSetKey, that fixes incorrect MatchSetKey values in the ContactCenter database. If incorrect values have already been loaded into the production database or definitive-match-config.xml has been changed, you can run this batch process to update the MatchSetKey of every contact in the database. Note: Update MatchSetKey processes all contact records in the system and can impact performance. Guidewire recommends that you run this process during off hours and that you run multiple threads of the worker process.</p> <p>The original bug concerned the MatchSetKey property of ABContact, which defines a unique index and match criteria. It is calculated automatically before every insert and update to ABContact. When you use staging tables to load a set of data, the MatchSetKey values are calculated automatically. However, as defined in definitive-match-config.xml, if one of the fields used for the keys is encrypted, but the MatchSetKey column is not encrypted, incorrect values are stored.</p> <p>To prevent this problem from occurring, do not run the encryption process on the staging tables. If you want to encrypt any fields that store MatchSetKey values, set the fields and the MatchSetKey to be encrypted. When the server starts, it encrypts the fields automatically.</p>
Contact Roles	
CLM-14607	Fixed an issue with the ExceptionConstraint in entityroleconstraints-config.xml, which was not implemented correctly. The old implementation allowed Contacts of any type if an ExceptionConstraint was included. The ExceptionConstraint now works as designed and allows only Contacts of the type designated in the main constraint or a ExceptionConstraint.
CLM-16241	Corrected an issue that caused a null pointer exception while searching from the Address Book tab for contacts that satisfy certain criteria.
CLM-16700	Modified ClaimContactRole to find existing contacts as appropriate. Previously, method addRole could potentially create a duplicate contact. Now, before saving, ClaimContactRole attempts to remove any duplicates, substituting the existing contact in place of the new duplicate.
Core	
CLM-12151	Guidewire has removed the GwXCostCat and GwXExtArray example entities in config/extensions (along with any references to them from other entities) as they are no longer used.

ID	Description
PL-14410	<p>Modified how ClaimCenter handles proximity searching:</p> <ul style="list-style-type: none"> Added a RadiusSearchMaxResults property to the ProximitySearchParameters non-persistent entity. This property indicates the maximum number of items to return on a radius search. ClaimCenter ignores this value if the value of DistanceBasedSearch is false. If the value of DistanceBasedSearch is true, and the value of RadiusSearchMaxResults is null, zero, or negative, then ClaimCenter limits the search by the value of ProximityRadiusSearchDefaultMaxResultCount. Added configuration parameter ProximityRadiusSearchDefaultMaxResultCount with a default of 1000. This parameter sets the maximum number of results that a proximity search is to return if using a fixed radius (n miles/KM). ClaimCenter ignores this parameter if performing an ordinal (nearest-n) search.
PL-15278	<p>Added a new optional attribute, forceEqMatchType, that can be added to the Criterion element in a search configuration file. If this attribute is present, it must be the name of a Boolean property on the criteria entity.</p> <ul style="list-style-type: none"> If this attribute evaluates to true, the Criterion uses an eq (equality) match. If this attribute evaluates to false, the Criterion uses the matchType specified in the Criterion. <p>For example:</p> <pre><Criterion property="StringProperty" forceEqMatchType="FlagProperty" matchType="startsWith"/></pre> <p>This Criterion uses a startsWith match for StringProperty <i>unless</i> the FlagProperty on the criteria entity is true, in which case the Criterion uses an eq match type.</p>
PL-15958	Fixed an issue in which stopping the global cache did not stop any currently running background thread(s) and actually started them if they were not already started.
PL-15959	<p>Corrected an issue with a null pointer in a global cache reaper thread that prevented the ClaimCenter application server running on WebSphere from shutting down gracefully. The issue caused the following error to occur:</p> <pre>ERROR Unexpected error while in global cache reaper thread: java.lang.RuntimeException: java.lang.NullPointerException</pre>
Database	
PL-12073	Added the ability to override the createhistogram attribute on the CreateTime column for a specific table. If you want to force the creation of a histogram on an base configuration column, then you can do so by adding a <histogramstatistics> element for the column in the appropriate <tablestatistics> element of the <databasestatisticselement> in config.xml.
PL-12200	<p>Modified the behavior of (and slightly renamed) the DBMS Counter Threshold (millis) option on the (Server Tools) Guidewire Profiler configuration page:</p> <ul style="list-style-type: none"> Name change - Renamed DBMS Counter Threshold (millis) to DBMS Instrumentation Capture Threshold for each Action (millis). Behavior change - Formerly, this option generated a DBMS report if any database operation exceeded a preset threshold value. Now, the profiler generates a DBMS report if the action (start of profiling session, end of profiling session) exceeds the threshold value set by this option. New option parameter Diff DBMS Counters - Set to capture the DBMS counters at the beginning of the profiling session and to include analysis of the differences in the DBMS-specific report. This option is meaningful only if you set the first option.
PL-15867	Corrected an issue that occurred while attempting to download database statistics if the database contained a column that was not in the base application data model and there was a histogram on the column. ClaimCenter now logs a warning and permits the download to continue.
PL-15873	<p>Guidewire now persists the results of consistency checks to the database for use of performance monitoring. You can monitor the progress of a particular consistency check run by using SQL queries against the associated tables, which are:</p> <ul style="list-style-type: none"> xx_dbconsistcheckrun xx_dbconsistcheckqueryexec <p>The prefix, xx, is the application prefix.</p> <p>Entity DBConsistCheckRun represents a run of the consistency checker. It contains an array of DBConsistCheckQueryExec entities, which represent the execution of an individual consistency check within a run.</p>

ID	Description
PL-15896	Added a (Server Tools) Info page called Consistency Check Runs . You use this page to download and view the results of database consistency checks. The page also provides an option to submit a batch job to asynchronously execute database consistency checks
PL-15898	Database consistency checks can now be run as a batch job, and the results downloaded and viewed, from the new Consistency Check Runs page accessed under Info Pages in Server Tools . You can configure the batch job to run checks on all or a subset of the database tables.
Financials	
CLM-12947	Fixed an issue in which ClaimCenter was improperly deleting the TransactionOffsetOnset entity during claim purge if the target claim of a transferred check was the only claim being purged.
CLM-13243	Added new integrity and consistency checks for CheckPortion, TransactionLineItem, and Deduction. For each entity, added checks for the following cases. <ol style="list-style-type: none"> 1. Reporting amount equals transaction amount if reporting currency equals transaction currency. 2. Transaction amount equals claim amount if transaction currency equals claim currency. 3. Claim amount equals reporting amount if claim currency equals reporting currency.
CLM-14909	Added a setOpenRecoveryReserves method to Claim for use in creating claim-level Recovery reserves.
CLM-15552	Fixed an issue in which the recoding of a final Payment in a multi-payment Check can cause Open Reserves to go negative.
CLM-15890	Fixed a performance issue in which having many reserves caused the Financial Summary screen to take too long to load.
CLM-15993	Fixed an issue with the AggLimitCalc batch process that caused it to fail with a syntax error in a SQL query
CLM-16095	Fixed an issue in which entering a money amount such as .23 generated a validation error. This is because ClaimCenter required that you enter 0.23 (with a leading zero) instead. Guidewire has modified fieldvalidators.xml so that 0 digits before the decimal point is legal.
CLM-16110	Fixed an issue that caused a user to not be able to edit the ReserveLine on a rejected payment.
CLM-16577	Fixed an issue in which ClaimCenter did not properly create manual exchange rates if a reserve was created to offset a stop/void of a check.
CLM-16622	Fixed an issue in which the createRecovery method on Claim and Exposure (used for creating Recovery transactions in rules) would not accept a negative amount for the new Recovery transaction amount.
CLM-16908	Fixed a problem in which any exception during handling of a response from ISO would cause the Guidewire ISOReceive servlet to send an HTTP 400 error code (Bad Request) back to ISO. ISO would then resend the response, resulting in an infinite loop that needed manual intervention to fix. Also included fix for a particular response parsing problem that caused an exception to be thrown, leading to the infinite loop described previously. If an ISO response contained multiple PropertyLossInfo aggregates, the ISOReceive servlet failed to populate a non-nullable field in ISOMatchReport. This led to a failure when that ISOMatchReport was committed.
Integration	
CLM-15472	Fixed an issue in which a claim generated from the Auto First and Final wizard would close an activity that required approval, without approving the activity. ClaimCenter now requires that you approve the activity before you can close the activity.
PL-14412	Fixed an issue in which ClaimCenter did not populate the description field properly as it created a document from a template if the description existed in the template descriptor.
PL-14627	Modified how the application generates domain graphs in graphic format. If you install a graphing program that can display DOT notation files, and add that program to your machine's PATH environment variable, then ClaimCenter can invoke your local graphing software. In that case, the download ZIP file will include a PDF version of the graphics file, in addition to the raw DOT file.
PL-14650	Improved the performance for XML generation of large Guidewire XML models.

ID	Description
PL-14960	<p>ClaimCenter now handles the encoding attribute on the XML header (the processing instruction). This attribute indicates the character encoding of the XML document. For example:</p> <pre><?xml version="1.0" encoding="UTF-8"?></pre> <p>For a discussion of XML character encoding, see the following:</p> <p>http://www.w3.org/International/0-charset</p> <p>Note: Guidewire supports the Java UTF-16BE and UTF-16LE characters sets as UTF-16 XML encoding, with the appropriate handling for byte-order bytes.</p>
PL-14971	<p>Modified <code>import.xsd</code> to add an <code>is-encrypted</code> attribute to the schema:</p> <pre><xs:attribute default="false" name="is-encrypted" type="xs:boolean"/></pre> <p>Use this to flag a field as encrypted, for example:</p> <pre><xs:attribute name="BankAccountNumber" type="gw:encoded-string" is-encrypted="true"/></pre>
PL-15327	<p>Added configuration parameter <code>ReCallDocumentContentSourceAfterRollback</code>. This parameter causes ClaimCenter to call the <code>isDocument</code> and <code>addDocument</code> methods again if there is a roll-back during a bundle commit.</p> <p>If a roll-back of the bundle does occur, ClaimCenter rolls back each entity in the bundle. The initial commit extracted the document content from the web request. The document content no longer exists after the bundle commit. Therefore, it is not possible to roll-back the document ID, even though it is possible to retrieve all other document fields. This can cause problems to occur.</p> <p>Guidewire recommends the following:</p> <ul style="list-style-type: none"> • If you have extended the Document entity, then set this parameter to <code>true</code>. • If you have not extended the Document entity, then it is most likely that you want to set this parameter to <code>false</code>. • If you are using a <code>IDocumentMetadataSource</code>, then set the parameter to <code>false</code>. If you then experience validation exceptions, test your set up, and, if necessary, set this value to <code>true</code>.
PL-15827	<p>Fixed an issue in which the Administration → Workflows → Manage Workflows page become confused and did not function properly if there were multiple versions of a workflow.</p>
Miscellaneous	
CLM-15527	<p>Fixed an issue in which the snapshot upgrade work queue (<code>com.guidewire.cc.system.crypt.SnapshotUpgradeWorkQueue</code> in the <code>work-queue.xml</code> file) threw a <code>ClassCastException</code> as it attempted to retrieve an entity type of an object snapshot in which the entity type did not exist. This could happen if an entity captured in the snapshot was removed in later versions of ClaimCenter. If the entity in the snapshot no longer exists, ClaimCenter generates a logging message that reads:</p> <pre>Snapshot upgrade skipping entity.X for claim (id=X) because the type cannot be found.</pre>
CLM-16542	<p>Fixed an issue with the Advanced Claim Search screen. You can use this screen to restrict your search to claims that have one or more critical indicators. There was an issue with this feature that meant that claims with no indicators at all (which can happen after upgrade or a staging table import) were included along with the claims that had the desired indicators.</p>
PL-16548	<p>Fixed an issue with multi-byte character display in PDF documents generated from a Guidewire template. This occurred if you attempted to generate a PDF document by performing the following sequence:</p> <ol style="list-style-type: none"> 1. Select a claim in ClaimCenter. 2. Navigate to Actions → New Document → Create from a template. 3. Browse to a template and Select Acrobat Sample. 4. Click Create Document. <p>The created PDF did not display field values that contain multi-byte characters unless you first selected the field in the PDF. If you deselected the field, the characters disappeared again.</p> <p>Note: All document templates will need to employ a font containing the full range of characters potentially exported to that template.</p>
Studio	
PL-10504	<p>Corrected a Studio issue in which debuggging in Browser mode resulted in an out-of-memory error.</p>
PL-10553	<p>Corrected an issue in which debugging a rule in Studio did not display and highlight the current line number in the source window.</p>

ID	Description
PL-13282	Corrected a display issue in which Studio truncated the error message that it showed if you hovered the mouse over a non-valid widget in the PCF editor.
PL-14174	Corrected an issue with the Studio auto-save feature that did not work correctly if you used CTRL-N to open another view before saving the view that was currently open.
PL-14367	Added a localization file import command to the command line system tools with the following syntax: <pre>import-110ns -Dimport.file=<translation file> -Dimport.locale=<destination locale></pre> This command provides the same functionality as that of the Studio Localizations → Display Keys (or Type-list Localization) right-click menu command Import Translation File .
PL-14518	Corrected an issue with the Studio rule debugger in which stepping through breakpoints set in rules did not update the Debugger pane (at the bottom of the screen) properly.
PL-15430	Modified Studio behavior so that an incorrect typelist or entity definition does not prevent Studio from starting. Instead, Studio logs the error in the Studio console. It is also possible to correct the error without requiring a restart of Studio.
Upgrade	
CLM-15977	Removed unnecessary <typecode1ocalization> nodes from the Currency typelist. These nodes caused difficulty during the Configuration Upgrade merge step.
Web	
PL-13053	Modified the way that ClaimCenter handles validation in the user interface. Checking, then unchecking an input group and then moving to another wizard step no longer triggers validation on the elements in the now hidden input group.
PL-13792	Corrected an interface issue in which copying and pasting a phone number that included an extension did not paste correctly into another field.
PL-14721	(CLM-14598) Fixed an issue that caused a 0-scale currency (for example, the Japanese Yen) to appear with a trailing decimal until the page was saved (or cancelled). This issue occurred if a currency-formatted input field was updated using reflection.
PL-15246	Corrected a button display problem with Internet Explorer 7 and Internet Explorer 8 that occurred if all of the following were true: <ul style="list-style-type: none"> • The button was disabled. • The header cell was blank or header text was shorter than the button.
PL-16532	Fixed a ClaimCenter security vulnerability that permitted a phishing attack through the manipulation of frames.

Known Issues and Limitations

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

- ClaimCenter Known Issues
- Platform/Studio Known Issues

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

Exception in the Bulk Invoice screen after restoring and reopening an archived claim (CLM-174)

Issue: If you archive a claim, restore it within 60 minutes of archiving it, and then immediately attempt to view a bulk invoice involving that claim, the system may show an exception message. This is because it takes time for the global cache to be updated after the claim is archived. However, this does not cause any data loss or corruption.

Workaround: You must wait 60 minutes after archiving a claim before restoring it. The length of time to wait is determined by the `GlobalCacheStaleTimeMinutes` configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt to view the bulk invoice for 60 minutes. Guidewire may not fix this issue, since restoring archived claims so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

Link issues between a new contact from the New Check wizard and ContactCenter (CLM-12172)

Issue: If you try to link a contact newly created from the New Check wizard to ContactCenter, the contact details can be overridden if the contact already exists in ContactCenter. This occurs if there is a match between the two integrated systems. Rather than creating a new contact, the existing matching contact information in ContactCenter overwrites the contact's information in ClaimCenter.

Workaround: Create the contact first in ClaimCenter (not in the New Check wizard), then if you want, link it to ContactCenter. Finally, use the New Check wizard to create a check for that contact.

Tax ID fields are not masked in claim snapshots that were created in earlier versions of ClaimCenter (CLM-12879)

Issue: Tax ID fields in the claim snapshot are not masked in the user interface. This happens because the claim snapshot is using an earlier version of the `ClaimSnapshotContactxxx.pcf` file which does not have the field encryption on the claim snapshot. (xxx represents various versions of the pcf file.)

Workaround: You must manually add a mask to the encrypted fields in earlier versions (such as 5.0.0) on the `ClaimSnapshotContactxxx.pcf` file.

Unable to generate a Claim Metrics report through the InetSoft Report Portal (CLM-13141)

Issue: The InetSoft onLoad script is designed to be run from within ClaimCenter and it should pass the report the locale. If the report is run outside of ClaimCenter, the locale needs to be set within InetSoft.

Workaround: Run reports through ClaimCenter.

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.

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"`.

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.

Localized document pattern specified in the activity pattern is not passed to the activity (CLM-13670)

Issue: When ClaimCenter is localized to the Japanese language, and you create an activity using the *Gosu Sample Email Sent Record* activity pattern, the system does not select the correct language.

Workaround: Use the document search feature to find the applicable localized document template.

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.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, then continue to use the deprecated methods.

Must regenerate database statistics after large batch processing of addresses (CLM-13963)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the ContactCenter database), you must run database statistics on the ab_abaddress table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in *filename*.

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)
```

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.

Duplicate descriptions for image/jpg mime types in the configuration file (CLM-14188)

Issue: In the file `modules\pl\config\locale\en_US\display.properties`, there are two entries under MIME types for jpeg file types that are defined with the same description, JPEG Image:

- `pjpeg`
- `jpeg`

These definitions cause a problem because the descriptions of both mime types are identical. If you have the **New Document** screen open to add a new image document and you click the **Document Type** drop-down list, you see two entries for **JPEG Image**. If you choose the wrong image type and add the new document, an error occurs.

Workaround: Open Guidewire Studio and do the following:

1. Navigate in the left pane to **Configuration** → **Display Keys** → **Mimetype** → **Configuration** → `image_pjpeg`.
2. Change the **Locale: English (US)** text to **Progressive JPEG Image**.
3. Save your changes and restart ClaimCenter.
4. Now in the ClaimCenter **New Document** screen, when you click the **Document Type** drop-down list, you see two choices for these file types, **JPEG Image** and **Progressive JPEG Image**.

No warning that you will overwrite a Service Provider Review or Question Set (CLM-14365)

Issue: If you do not retire the review or question set and import the new review or question set, then there is no warning or mechanism to prevent you from overwriting the existing review or question set.

Workaround: You should create the new review or question set in a new `.csv` file. Also, ensure that all public IDs of all entities of the new review or question set are unique and different from the existing review or question set. Consider that any entities such as `ReviewType`, `ReviewCategoryQuestionSet`, `QuestionSet`, `Question`, `QuesitonChoice`, and `QuestionFilter` could be overwritten.

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.

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.

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.

UnsupportedOperationException: MetaIntrinsicType does not support array yet (CLM-15718)

Issue: ClaimCenter 3.x snapshot PCF files generate a MetaIntrinsicType does not support array yet exception if the snapshot PCF contains the following cell:

```
Cell id="Lienholders" value="util.Snapshot.renderValue(PolicyVehicle.Vehicle.Lienholders)"
  label="displaykey.Web.ClaimSnapshotPolicyVehicles300LV.Lienholders"/>
```

You see this problem if you navigate to the **FNOL Snapshot** → **Policy** → **Vehicle** screen and select a ClaimCenter 3.x snapshot claim that contains this particular cell.

Workaround: Navigate to the following Studio location and comment out the cell in any PCF file that contains it.

Page Configuration (PCF) → claim → snapshot → 300

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 before returning the archived claim document to be restored.

Existence of table cc_tmpcheckrpt can cause upgrade to fail (CLM-16668)

Issue: If you ran the FinancialsCalc batch process in a pre-5.0.8 version of ClaimCenter, it is possible that it populated a temporary table, cc_tmpcheckrpt. The 6.0.5 upgrade process does not update this table successfully,

which causes the upgrade to fail. The FinancialsCalc batch process does not run automatically in the base configuration. You must trigger it manually through the maintenance tools. For example:

```
maintenance_tool financialscalculations
```

Workaround: Delete table `cc_tmpcheckrpt` before upgrading. The batch process only uses the contents of this table while it is actively running. However, it does not remove the contents of the table after it completes. You can safely delete this table.

Statistics batch process queries do not scale on large Oracle databases (CLM-16830)

Issue: Guidewire has determined that some of the queries used by the dashboard statistics batch process (DashboardStatistics) do not scale well on large Oracle databases. The dashboard statistics batch process can take up to 14 hours on a 5 million claim multi-byte database. This is problematic, even though the statistics batch process runs just once a night and other tasks appear to complete normally while it is running.

Workaround: It is possible to improve performance by doing the following:

1. Alter an index on Transaction to include more columns, which reduces table accesses by the dashboard statistics queries.
2. For Oracle, use Oracle Stored Outlines to force a better plan for some of the dashboard statistics queries.

To review details of these performance enhancements:

1. Navigate to the following link on the Resource Portal:
<http://guidewire.custhelp.com/app/resources/infrastructure/documents>
2. Click **Infrastructure** under **Document Resources**.
3. Select the following white paper from the list under **Guidewire Infrastructure Whitepapers**:
Guidewire ClaimCenter 6.0 document for Oracle.pdf

Platform/Studio Known Issues

First time you click on the arrow of the typekey input, the drop-down menu will not open (PL-10134)

Issue: The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting `InputHelpTextOnFocus` to `false` in the `config.xml` file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

Studio Rules do not use correct capitalization for root object's name (PL-10740)

Issue: Rule set root objects are not named with first letter lower-cased.

Workaround: Rules engine will issue warnings when correct case for objects is not being used.

Studio—with Subversion (SVN)—incorrectly copies `.svn` files (PL-10932)

Issue: *This issue affects installations that use Subversion only.* If you modify the base configuration rules in Studio, Studio creates a copy of the rule in the `modules/configuration` folder. Studio then executes Subversion commands to add the files and folders to Subversion. In the process, Studio incorrectly copies the `.svn` folder and its children. This leads to Subversion problems.

Workaround: Manually delete the copied `.svn` folders.

Geocode plugin calls *isSufficientlyCompleteToGeocode* method twice for each address (PL-11578)

Issue: The Geocode plugin calls the `isSufficientlyCompleteToGeocode` method twice for each address.

Workaround: Guidewire is aware of this issue.

Countries configured in `zone-config.xml` still generate a warning during regen-dictionary even when zone data is loaded for all of these countries (PL-11947)

Issue: Countries configured in `zone-config.xml` still generate a warning during regen-dictionary even when zone data is loaded for all of these countries.

Workaround: Warning message is created in error and can safely be ignored.

User interface cannot handle starting multiple instances of a batch process (PL-12372)

Issue: The user interface cannot handle multiple instances of a batch process.

Workaround: If you need to execute multiple instances of a batch process, then you need to start each from the command line. Also, you need to ensure that the `BatchProcess.isExclusive` method returns `false` to allow the application to run multiple instances simultaneously.

Type system refresh after PCF page title change does not update corresponding menu label (PL-13057)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

Length limitation on entity localization table names (PL-13360)

Issue: There is a length limitation on entity localization table names.

Workaround: Ensure that localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

US-Locations.txt file with the U.S. geodata from GreatData has special characters that cause validation problems with United States Postal Service (USPS) data (PL-13384)

Issue: The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can process the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

Use of *StringBuilder* class prevents debugging of web service (PL-13435)

Issue: Suppose that you have a web service in which you have a method with a parameter of type `X` and inside the web service method you reference a Gosu class of type `Y`. This can be an issue if the Gosu class `Y` has a `toString` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you attempt to set a breakpoint in the web service method and try to step through the method, Studio generates errors.

Workaround: Studio does not generate errors if you call the web service in normal run mode rather than in debug mode.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

Email with file attachment with unicode filename not correctly handed over to the mail server (PL-13582)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

The GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

JavaToolkit.gs has incorrectly hard coded memory which results in failed *regen-java-api* Ant task (PL-13663)

Issue: The *JavaToolkit.gs* has hard coded memory which can result in failed *regen-java-api* Ant tasks.

Workaround: Increase the size of the maximum heap setting on line 161 of *JavaToolkit.gs* in the Ant module. The default value is 512.

Cannot make a delegate field a localized column (PL-13761)

Issue: You cannot make a delegate field a localized column.

Workaround: Move the column that you want to localize off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define an enhancement property on the delegate, that *delegates* to the appropriate column, depending on the implementing entity.

Cannot deploy custom *web.xml* file (PL-14482)

Issue: ClaimCenter does not deploy a custom *web.xml* file properly.

Workaround: Before attempting to deploy the application, do the following:

1. Open the following file for editing:

```
ClaimCenter\modules\ant\public\gw\ant\deploy.gs
```

2. Search for the following lines of code:

```
var deployDir = module.Dir.file("deploy")
if (deployDir.exists()) {
  new Copy() { :FileSet = deployDir.fileSet(), :ToDir = destDir }
    .execute()
}
```

3. Make the following change:

```
var deployDir = module.Dir.file("deploy")
if (deployDir.exists()) {
  new Copy() { :FileSet = deployDir.fileSet(), :ToDir = destDir, :Overwrite = true }
    .execute()
}
```

Notice that you need to add `:Overwrite = true` to the Copy method.

Cannot print second-level *ListView* (PL-14640)

Issue: Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all of the items.

Workaround: Remove the lower level list view out of the hierarchical structure and include it at the top level.

Superuser role unable to edit or delete calendar holidays (PL-14942)

Issue: It is not possible to edit or delete a calendar holiday as superuser.

Workaround: Log into ClaimCenter using su to be able to edit the holiday schedule.

RowSet and *RowIterator* configured in the same *ListView* do not work correctly. (PL-14946)

Issue: Configuring a *RowSet* and *RowIterator* in the same *ListView* does not provide the right behavior for editability.

Workaround: Separate the *RowSet* and *RowIterator* that exist in the same *ListView* into separate components or into multiple *ListView* widgets.

Masked input fields do not correctly handle ENTER keypress (PL-14955)

Issue: Input fields that have a field validator mask do not correctly handle the ENTER keypress. After you press ENTER, the field no longer displays the input mask and it is not possible to enter any further input.

Workaround: Cancel the page and re-enter the information.

ClaimCenter does not recognize modified *Calendar.js* file (PL-14997)

Issue: ClaimCenter does not correctly recognize changes made to *Calendar.js* through Studio. You access this file in the following Studio location:

Resources → Web Resources → resources → javascript → global

Workaround: Instead, modify *global.js* directly. You access this file in the following installation file location:

ClaimCenter/webapps/cc/resources/javascript

WARNING This is one of the very few occasions in which you modify an application file outside of the application configuration folder. Ensure that you modify *global.js* only. Otherwise, you can invalidate your installation.

Studio debugger cannot debug *gw.sampledata* classes (PL-15357)

Issue: Attempting to debug a test that loads or uses base application sample data causes an unimplemented byte-code error in the JVM. This means that you cannot use the Studio debugger to debug *gw.sampledata* classes.

Workaround: Guidewire is aware of this issue.

Batch process fails to purge claims (PL-16274)

Issue: Failure to purge one claim in the *BulkPurge* batch process (referred to as *purge* or *claimpurge* in the maintenance tools) can prevent the batch process from purging any other claims. As long as the problem with the claim persists, it will prevent other claims from being purged. The console log indicates this problem by a message similar to the following:

Exception thrown by batchprocess: BulkPurge

Workaround: Failures to purge a claim are usually caused by problems in customer extensions to the claim graph. Diagnose and correct the problems reported in the server console, then retry the batch process.

Exporting Admin data and reimporting causes issues (PL-16475)

Issue: Attempting to export Admin data from an upgraded version of ClaimCenter into a clean base copy of the same application version causes a number of issues.

Workaround: See Guidewire Knowledge Base article 1118 (*Exported admin data from upgraded cc602 is not importable into base cc602*). You can access Guidewire Knowledge Base at the following location:

<http://guidewire.custhelp.com/app/home>

You must have a valid account to access the Guidewire Resource Portal.

Configuration Upgrade Tool does not copy XSD files correctly (PL-16754)

Issue: As part of the upgrade from ClaimCenter 5.0.x to ClaimCenter 6.0.x, Guidewire requires that you run the Configuration Upgrade Tool, using the following command:

```
ant -f upgrade.xml upgrade
```

This script works incorrectly and does not copy various XSD files to the correct target directory properly.

Workaround: After running the Configuration Upgrade Tool automated steps, copy all XSD files from the following ClaimCenter 5.0.x directory:

```
...\configuration\config\registry\
```

to the following ClaimCenter 6.0.x directory:

```
...\modules\configuration\gsrc\
```

chapter 39

Guidewire ClaimCenter 6.0.6 Release Notes

Release 6.0.6

Copyright © 2001-2013 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Live, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

This material is Guidewire proprietary and confidential. The contents of this material, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

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

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

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
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 6.0.6.

For Standard Reporting, this release supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter
- Prior ClaimCenter release notes for any versions that have been skipped

The following issues pertain to installing this release.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-72C-ERX-0000BF00100001F-F2B025280327

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide* documentation.

The build number of the reporting tool must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Support

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

- At the Guidewire Resource Center – <http://guidewire.custhelp.com>
- By email – support@guidewire.com
- By phone – +1-650-356-4955

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Rules Changes
- Improvements to Upgrade Diff Report
- Improvements and General Issues
- Known Issues and Limitations

Base PCF File Changes

All links below require the ReleaseNotes_files directory on your local disk.

ClaimCenter release 6.0.5 to 6.0.6

- 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/pl directory, [click here](#).

Rules Changes

ClaimCenter release 6.0.5 to 6.0.6

- There are no changes to the base rules in the modules/cc directory.

Improvements to Upgrade Diff Report

With previous releases, Guidewire provided a static report in the release notes detailing certain differences between the current release and the prior minor release. This report described changes in display keys, entities, typelists, and the Gosu API. Guidewire has improved this report to dynamically conform to each customer's particular upgrade path. In addition, the report is regularly updated as the tools for generating it are enhanced to provide more information.

Because the new report is tailored to your particular upgrade requirements, it is no longer included with the general release notes. To obtain your custom Upgrade Diff Report, contact your Guidewire representative. In the future, this report will be available on the Guidewire Resource Center web portal, allowing you to view the latest version right when you need it.

Improvements and General Issues

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

ID	Description
Administration, Catastrophe, Claim Associations, Dashboard, Desktop, Team, Question Sets	
CLM-1772	Previously, each time the application server started, ClaimCenter pulled the As of: date field on the Team Summary page from the most recent date in the UserGroupStats.CalculateDate field. If there were no records in this table, then ClaimCenter arbitrarily assigned the As of: date to 2/1/2003. Now, if UserGroupStats is empty and the server has just been rebooted, ClaimCenter displays a message informing the user that the Statistical Calculator has never been run.
Batch Processes	
PL-14589	Guidewire has implemented a new batch process, ProcessCompletionMonitor. The process wakes up at user-configurable intervals and examines the Process History and Work Queue for all batch processes. After the batch process completes, it invokes plugin IBatchCompletedNotification to permit user code to react to the completion of the batch process. This process also marks ProcessHistory.NOTIFICATIONSENT = true for that batch process to ensure that the plugin is invoked only once. Note that you must provide an implementation of plugin IBatchCompleteNotification.
Claim - FNOL Wizard, New Claim Wizard, Claim Snapshot, FNOL mapper, Language Pack	
CLM-16243	Guidewire now provides tool-tips for all icons in the ClaimCenter interface that are not accompanied by text that describes their purpose or function.
Claim Metrics	
CLM-17641	Corrected an issue in the Claim Health Metric screen involving the calculation % Reserve Change from Initial Reserve where the cumulative reserves for a claim are divided by the initial reserve amount. This failed if you added 10 times more reserves than what was initially set (which is 1000%, exceeding the precision of three digits). The issue was the PercentageMetric datatype. In the default configuration, ClaimCenter has a scale/precision of (3,0). This modification changed the PercentageMetric datatype to have a precision of 8 and scale 0.

ID	Description
Configuration	
PL-13071	Added the ability to verify PCF files outside of Guidewire Studio through the use of the <code>gwcc verify-types</code> command line utility.
ContactCenter	
CLM-17933	See "Integration between ClaimCenter 6.0.6 and ContactManager not working (CLM-17933)" on page 605
Contacts, Roles, SOAP APIs, Plugins, Rules - Core	
CLM-17800	Fixed a problem with the <code><IgnoreProperty></code> element in <code>contact-sync-config.xml</code> wherein the property was being ignored during Address Book synchronization for all contact subtypes rather than only the contact subtype for which the element was defined.
Database	
PL-16620	<p>Database parameter <code>checker.threads</code> (in <code>config.xml</code>) controls the number of threads used to execute database consistency checks (DBCC). Previously, you could only control the number of threads using the synchronous methods of running DBCC, from the <code>system_tools -checkdbconsistency</code> command line and at application start-up. With this change, you can override the number of threads if you are running consistency checks as a batch job.</p> <ul style="list-style-type: none"> If you set a value for <code>checker.threads</code> in <code>config.xml</code>, then ClaimCenter displays the configured number of threads on the (System Tools) Info Page → Consistency Checks page. You can override the configured value with a positive integer and submit new consistency checks with this new value on that page. If you do not set a value for <code>checker.threads</code> in <code>config.xml</code>, then ClaimCenter uses 1 as the default value and displays that value on the Consistency Checks page. You can still override this value on that page.
PL-16762	Oracle. Guidewire has added a new <code><database></code> parameter in <code>config.xml</code> that you use—in Oracle-based implementations—to disable rewrite queries using materialized views at a session level. The name of the parameter is <code>queryRewriteEnabled</code> . Set its value to <code>false</code> to disable rewrite queries.
PL-16763	<p>Oracle. Added a new configuration parameter—<code>QueryRewriteForClaimSearch</code>—that enables options for query re-write on Oracle. There are options to rewrite queries using materialized view based on cost or to force the rewrite. The four possible values are as follows:</p> <ul style="list-style-type: none"> <code><param name="QueryRewriteForClaimSearch" value="COST/NOSTALE"/></code> <code><param name="QueryRewriteForClaimSearch" value="COST/STALE"/></code> <code><param name="QueryRewriteForClaimSearch" value="FORCE/NOSTALE"/></code> <code><param name="QueryRewriteForClaimSearch" value="FORCE/STALE"/></code> <p>If you provide an invalid value, the server ignores it.</p>
PL-17133	Fixed an issue in which ClaimCenter would open a database connection and never close it if accessing an external data source using JNDI. This prevented database fail-over. This fix changes this behavior so ClaimCenter closes all connections if they are no longer in use.
Document Management	
CLM-17262	See "Method <code>adjustDocumentName</code> can cause an infinite loop (CLM-17262)" on page 605.
Financials	
CLM-16833	Fixed an issue in the Check wizard in which the detection of a possible duplicate check would result in the <code>ReportableAmount</code> being <code>null</code> , and deductions not being properly computed.
CLM-16939	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.
CLM-16969	<p>Added methods on <code>ReserveSet</code> that enable more flexible creation of reserves transactions in Gosu business rules. The new methods are:</p> <ul style="list-style-type: none"> <code>newReserve</code> <code>prepareForCommit</code>
CLM-17088	Fixed an issue in the <code>addItem(ID, items)</code> Bulk Invoice API that prevented ClaimCenter from setting an exposure on Bulk Invoice Items.
CLM-17146	Removed restriction on manual checks that required them to be positive, allowing the user to create negative ones.
CLM-17506	Fixed an issue in which the <code>Payment.acknowledgeRecode</code> and <code>Check.acknowledgeTransfer</code> methods were not moving onset transactions to <i>Submitted</i> status.

ID	Description
CLM-17613	<p>Added methods to CheckCreator to allow much more flexibility if creating new Check objects in Gosu business rules. This allows you to modify the new check however you want—such as adding additional payments or line items—before submitting for approval and updating TAccount objects. The new methods are:</p> <ul style="list-style-type: none"> • createCheck • prepareForCommit • createAndPrepareForCommit
CLM-17759	Fixed an issue in the PaymentExceedsReserves (PER) authority limit, where evaluation of a supervisor's Authority Limits would seem to double count the amount of the payment.
CLM-17763	<p>Fixed an issue in which ClaimCenter could potentially throw an exception if transferring a multicurrency check for a BulkInvoiceItem from one claim to another, and, if 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.</p> <p>Now, instead, ClaimCenter copies the Deduction's ReportingAmount to the ClaimAmount. This is done because the claim and reporting currencies are equal and so these amounts must be equal. Guidewire also modified CheckTransferDV.pcf 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.</p>
Geocoding	
PL-13499	<p>Guidewire has added an implementation of the Geocode plugin that connects to the Microsoft Bing Maps Geocode Service. The Bing Maps plugin implementation replaces the Microsoft MapPoint implementation.</p> <p>Guidewire deprecates the MapPoint implementation with this release. 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 before November 18, 2011. Otherwise, geocoding features in the application cease to function on November 18, 2011, and afterwards.</p>
Infrastructure	
CLM-15974	Previously, if you set a large number of days on the Metrics & Thresholds page, it caused new claims to either complete after many hours or to hang and never advance to completion. The Metrics & Thresholds page now contains a Maximum Number of Metrics Days field. An administrator can set this number to limit metrics days to an optimal number that will not cause claims processing to delay or to fail.
Integration	
CLM-17555	Added Metropolitan Report Requests with the same Report Type, Date of Loss, Insured Name, and Claim Number as previously requested reports. Reports no longer get stuck in the Metropolitan Report workflow because of their 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-17707	Previously, the MetroReport.PastWorkflowTimeout field would change to true. This would close the Metropolitan Report workflow and generate an error activity for all open metropolitan reports. Now, it is possible to set Metro.WorkflowTimeout to more than several years without open metropolitan reports having their status prematurely set to Closed.
CLM-17933	See “Integration between ClaimCenter 6.0.6 and ContactManager not working (CLM-17933)” on page 605
Miscellaneous	
CLM-17148	<p>Previously, while performing Statistical Calculations run in a clustered environment, ClaimCenter populated the Team Summary As of: field from the ProcessHistory.CompleteDate field. However, this field was out of synchronization with the actual completion date on the batch process server. This is because ClaimCenter updated the ProcessHistory table after the call was made to update the Team Summary page.</p> <p>Now, the Statistics Calculator completion date is passed directly from the batch process server to the other ClaimCenter servers in the cluster. The ClaimCenter server then updates the Team Summary As of: field.</p>

ID	Description
CLM-17682	<p>ClaimCenter 6.0.6 now uses ISO v5.3 for Match Reporting.</p> <p>The following new elements can occur under <code>com.iso_CovInfo1</code> element in <code>com.iso_AddCovInfo</code>:</p> <ul style="list-style-type: none"> <code>com.iso_DeleteFromCMS</code> (IXMLNode) <code>com.iso_NotSendCovCMS</code> (IXMLNode) <p>The following new element can occur under <code>xsd.iso.ClaimInvestigationAddRq</code>:</p> <ul style="list-style-type: none"> <code>com.iso_RecallRqInd</code> (IXMLNode) <p>The <code>ClaimsOccurrence</code> element under <code>ClaimInvestigationAddRq</code> now has an array of <code>ProbableIncurredAmt</code> elements called <code>ProbableIncurredAmts</code>.</p> <ul style="list-style-type: none"> The <code>ProbableIncurredAmt</code> contains an optional currency. It was previously limited to a single occurrence. <p><code>SuitFiledInds</code> and <code>ClosedDts</code> are now arrays on <code>ClaimsPartyInfo</code>. They were previously limited to a single occurrence.</p> <p>Also new:</p> <ul style="list-style-type: none"> <code>com.iso_TheftTypeInd</code> is a new String type under <code>InvestigationInfo</code>. <code>com.iso_VehRecalls</code> is a new array of <code>com.iso_VehRecall</code> instances (of type IXMLNode) under <code>InvestigationInfo</code>. <code>com.iso_SFNMTISs</code> is a new array of <code>com.iso_SFNMTIS</code> instances (of type IXMLNode) under <code>InvestigationInfo</code>. <code>com.iso_DeleteFromCMS</code> is a new IXMLNode under <code>com.iso_CovInfo1</code> element under <code>com.iso_AddCovInfo</code> under <code>xsd.iso.ClaimInvestigationAddRq</code>. <code>com.iso_StopCMSQuery</code> is a new IXMLNode under <code>com.iso_SIUParty</code>.
Performance	
CLM-17050	<p>Corrected an issue with missing and incorrect indexes on <code>ClaimInfo</code> and related entities that caused performance problems with archived claim search. Performance testing showed that base configuration indexes provided in the ClaimCenter 6.0.5 release were inadequate for archived claim searches. Thus, Guidewire has modified and added indexes on <code>ContactInfo</code> and <code>ClaimInfoAccess</code> in ClaimCenter 6.0.6.</p> <p>The new indexes significantly improve the performance of searches for archived claims:</p> <ul style="list-style-type: none"> Customers who have not customized indexes on these entities do not need to do anything, the database upgrade automatically modifies and adds the new indexes. Customers who have customized indexes on these entities need to look at the new indexes provided in ClaimCenter 6.0.6 and decide whether to use them. If the new indexes do not add anything to the existing customized indexes you remove them using <code>remove-index</code> in the entity extension file.
Queries	
PL-12485	<p>Guidewire has deprecated the <code>RawQuery</code> property. Instead, create a <code>gw.api.filters.StandardQueryFilter</code> and use its <code>filterQuery</code> method or the <code>IQueryBeanResult.addFilter</code> method to apply the filter to a query.</p>
Search	

ID	Description
CLM-16963	<p>Modified archived claim search so that you can now configure it in a similar fashion to the standard claim search. If, on the Advanced Search screen, you select Archive to search for archived claim, then ClaimCenter runs the subsequent search against the <code>ClaimInfo</code> and <code>ContactInfo</code> tables instead of the standard <code>Claim</code>, <code>ClaimContact</code>, <code>Contact</code> (and other) tables. In previous ClaimCenter versions, there was no way of customizing the search against the <code>ContactInfo</code> table, which contains information about the insured and claimant contacts for archived claims.</p> <p>This fix allows you to modify the following section to configure such searches:</p> <pre><!-- Mapping from CCNameCriteria to ContactInfo, used if searching for archived claims --> <CriteriaDef entity="CCNameCriteria" targetEntity="ContactInfo"> <Criterion property="CompanyName" targetProperty="Name" matchType="startsWith"/> <Criterion property="FirstName" matchType="startsWith"/> <Criterion property="LastName" matchType="startsWith"/> <Criterion property="TaxId" targetProperty="TaxID" matchType="eq"/> </CriteriaDef></pre> <p>The default configuration, as shown, uses <code>startsWith</code> matches on first name, last name and company name, and equality matching for <code>TaxID</code>. Customers can add their own extension fields to <code>Contact</code> and, providing they add matching extension fields to <code>ContactInfo</code>, ClaimCenter copies these fields to <code>ContactInfo</code> if ClaimCenter archives the corresponding claim. Customers can then add <code>Criterion</code> definitions for searching these extension fields in the standard way.</p>
CLM-17075	<p>Removed fields from the <code>ClaimInfoSearchView</code> view entity as ClaimCenter only uses this view entity to display information about archived claims for which these fields are nearly always null. These fields include:</p> <ul style="list-style-type: none"> • <code>ActiveAdjuster</code> • <code>ActiveInsured</code> • <code>ClaimCurrency</code> • <code>ClaimNumberOrder</code> • <code>ClaimStateOrder</code> • <code>Flagged</code> • <code>FuturePayments</code> • <code>LossDateOrder</code> • <code>PolicyNumberOrder</code> • <code>RemainingReserves</code> • <code>TotalPayments</code> <p>Guidewire has also modified the <code>ClaimSearchResultsLV.pcf</code> and <code>ArchiveClaimRetrievePopup.pcf</code> files to no longer display these fields. All display keys used to reference these fields in the user interface have also been removed.</p>
Upgrade	
PL-18021	Fixed an issue in which the Upgrade Tool did not handle rules upgrade properly
Web	
PL-16711	Improved performance by modifying the UI-rendering JavaScript so that it now only resizes visible widgets.
XML	
PL-16706	Update to allow multi-threading of date formatting for File Based Archiving to improve performance.

Known Issues and Limitations

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

- ClaimCenter Known Issues
- Platform/Studio Known Issues

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

Integration between ClaimCenter 6.0.6 and PolicyCenter 7.0.2 does not work (PC-15581)

Issue – The PolicyCenter 7.0.2 release does not contain a required RPC version of the ClaimToPolicySystemNotificationAPI web service that ClaimCenter 6.0.6 needs for integration with PolicyCenter 7.0.2.

Workaround – Contact Guidewire Support for the proper version of this API.

Exception in the Bulk Invoice screen after restoring and reopening an archived claim (CLM-174)

Issue: If you archive a claim, restore it within 60 minutes of archiving it, and then immediately attempt to view a bulk invoice involving that claim, the system may show an exception message. This is because it takes time for the global cache to be updated after the claim is archived. However, this does not cause any data loss or corruption.

Workaround: You must wait 60 minutes after archiving a claim before restoring it. The length of time to wait is determined by the GlobalCacheStaleTimeMinutes configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt to view the bulk invoice for 60 minutes. Guidewire may not fix this issue, since restoring archived claims so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

Link issues between a new contact from the New Check wizard and ContactCenter (CLM-12172)

Issue: If you try to link a contact newly created from the New Check wizard to ContactCenter, the contact details can be overridden if the contact already exists in ContactCenter. This occurs if there is a match between the two integrated systems. Rather than creating a new contact, the existing matching contact information in ContactCenter overwrites the contact's information in ClaimCenter.

Workaround: Create the contact first in ClaimCenter (not in the New Check wizard), then if you want, link it to ContactCenter. Finally, use the New Check wizard to create a check for that contact.

Tax ID fields are not masked in claim snapshots that were created in earlier versions of ClaimCenter (CLM-12879)

Issue: Tax ID fields in the claim snapshot are not masked in the user interface. This happens because the claim snapshot is using an earlier version of the ClaimSnapshotContactxxx.pcf file which does not have the field encryption on the claim snapshot. (xxx represents various versions of the pcf file.)

Workaround: You must manually add a mask to the encrypted fields in earlier versions (such as 5.0.0) on the ClaimSnapshotContactxxx.pcf file.

Unable to generate a Claim Metrics report through the InetSoft Report Portal (CLM-13141)

Issue: The InetSoft onLoad script is designed to be run from within ClaimCenter and it should pass the report the locale. If the report is run outside of ClaimCenter, the locale needs to be set within InetSoft.

Workaround: Run reports through ClaimCenter.

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.

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"`.

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.

Localized document pattern specified in the activity pattern is not passed to the activity (CLM-13670)

Issue: When ClaimCenter is localized to the Japanese language, and you create an activity using the *Gosu Sample Email Sent Record* activity pattern, the system does not select the correct language.

Workaround: Use the document search feature to find the applicable localized document template.

'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`.

Workaround: Use destination IDs instead of message sink IDs if you can. If this is not possible, then continue to use the deprecated methods.

Must regenerate database statistics after large batch processing of addresses (CLM-13963)

Issue: After geocoding a large batch of addresses (defined by new addresses greater than 10% of the current addresses in the `ContactCenter` database), you must run database statistics on the `ab_abaddress` table. This will help the database understand the distribution of the addresses and thus improve query performance.

Workaround: To generate a file containing the database commands, run the following:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

The database administrator should then run the commands contained in `filename`.

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)
```

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()) {
    NewClaimDuplicationsWorksheet.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()) {
    NewClaimDuplicationsWorksheet.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.

Duplicate descriptions for image/jpg mime types in the configuration file (CLM-14188)

Issue: In the file `modules\pl\config\locale\en_US\display.properties`, there are two entries under MIME types for jpeg file types that are defined with the same description, **JPEG Image**:

- `pjpeg`
- `jpeg`

These definitions cause a problem because the descriptions of both mime types are identical. If you have the **New Document** screen open to add a new image document and you click the **Document Type** drop-down list, you see two entries for **JPEG Image**. If you choose the wrong image type and add the new document, an error occurs.

Workaround: Open Guidewire Studio and do the following:

1. Navigate in the left pane to **Configuration** → **Display Keys** → **Mimetype** → **Configuration** → **image_pjpeg**.
2. Change the **Locale: English (US)** text to **Progressive JPEG Image**.
3. Save your changes and restart ClaimCenter.
4. Now in the ClaimCenter **New Document** screen, when you click the **Document Type** drop-down list, you see two choices for these file types, **JPEG Image** and **Progressive JPEG Image**.

No warning that you will overwrite a Service Provider Review or Question Set (CLM-14365)

Issue: If you do not retire the review or question set and import the new review or question set, then there is no warning or mechanism to prevent you from overwriting the existing review or question set.

Workaround: You should create the new review or question set in a new .csv file. Also, ensure that all public IDs of all entities of the new review or question set are unique and different from the existing review or question set. Consider that any entities such as ReviewType, ReviewCategoryQuestionSet, QuestionSet, Question, QuesitonChoice, and QuestionFilter could be overwritten.

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.

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.

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.

UnspportedOperationException: MetaIntrinsicType does not support array yet (CLM-15718)

Issue: ClaimCenter 3.x snapshot PCF files generate a MetaIntrinsicType does not support array yet exception if the snapshot PCF contains the following cell:

```
Cell id="Lienholders" value="util.Snapshot.renderValue(PolicyVehicle.Vehicle.Lienholders)"
  label="displaykey.Web.ClaimSnapshotPolicyVehicles300LV.Lienholders"/>
```

You see this problem if you navigate to the **FNOL Snapshot** → **Policy** → **Vehicle** screen and select a ClaimCenter 3.x snapshot claim that contains this particular cell.

Workaround: Navigate to the following Studio location and comment out the cell in any PCF file that contains it.

Page Configuration (PCF) → claim → snapshot → 300

ClaimCenter does not prevent concurrent restore of archived claim by second user (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` exception before returning the archived claim document to be restored.

Existence of table 'cc_tmpcheckrpt' can cause upgrade to fail (CLM-16668)

Issue: If you ran the `FinancialsCalc` batch process in a pre-5.0.8 version of ClaimCenter, it is possible that it populated a temporary table, `cc_tmpcheckrpt`. The 6.0.5 upgrade process does not update this table successfully, which causes the upgrade to fail. The `FinancialsCalc` batch process does not run automatically in the base configuration. You must trigger it manually through the maintenance tools. For example:

```
maintenance_tool financialscalculations
```

Workaround: Delete table `cc_tmpcheckrpt` before upgrading. The batch process uses the contents of this table only while it is actively running. However, it does not remove the contents of the table after it completes. You can safely delete this table.

Unable to archive a claim after performing policy refresh on it (CLM-16708)

Issue – If you attempt to archive a claim after performing a policy refresh on it, the archive attempt fails. The policy refresh process does not detect and sever all connections between the old policy graph and the claim, and these connections prevent the claim from being archived. If you wish to implement both claim archiving and policy refresh, then you need to sever these connections after policy refresh has run.

Workaround – In Guidewire Studio, refer to the new class `RetiredPolicyGraphDisconnecterExample` for an example of the type of code to execute after policy refresh has run to sever these connections. You can also refer to Knowledge Base article #1555 for more information.

Statistics batch process queries do not scale on large Oracle databases (CLM-16830)

Issue: Guidewire has determined that some of the queries used by the dashboard statistics batch process (`DashboardStatistics`) do not scale well on large Oracle databases. The dashboard statistics batch process can take up to 14 hours on a 5 million claim multi-byte database. This is problematic, even though the statistics batch process runs just once a night and other tasks appear to complete normally while it is running.

Workaround: It is possible to improve performance by doing the following:

1. Alter an index on `Transaction` to include more columns, which reduces table accesses by the dashboard statistics queries.
2. For Oracle, use Oracle Stored Outlines to force a better plan for some of the dashboard statistics queries.

To review details of these performance enhancements:

1. Navigate to the following link on the Resource Portal:
<http://guidewire.custhelp.com/app/resources/infrastructure/documents>
2. Click **Infrastructure** under **Document Resources**.
3. Select the following white paper from the list under **Guidewire Infrastructure Whitepapers**:
Guidewire ClaimCenter 6.0 document for Oracle.pdf

Method adjustDocumentName can cause an infinite loop (CLM-17262)

Issue – Previously, ClaimCenter used `DocumentProduction.adjustDocumentName` to append (1), (2),... (n) to any documents that have the same name as a document that already exists in ClaimCenter. This method works without issue if the document name is the key used to uniquely identify the document. However, if the document ID is employed instead, the system goes into an infinite loop.

To help remedy this, Guidewire has moved method `adjustDocumentName` from `DocumentProduction` to `DocumentEnhancement` to make it possible for you to remove or edit this value as needed.

Workaround – Modify `DocumentEnhancement` to remove the logic that performs the document rename. (The logic exists in a `while` loop.)

Integration between ClaimCenter 6.0.6 and ContactManager not working (CLM-17933)

Issue – ClaimCenter does not properly handle the integration between ClaimCenter 6.0.6 and ContactManager 7.0.0.

Workaround – Add the following to file `ab-to-cc-data-mapping.xml` in ClaimCenter 6.0.6:

1. Add the following:

```
<EntityMapping source="ABContactTag" target="ContactTag">
  <FieldMapping source="ABContact" mapperClassName="gw.api.util.mapping.NullFieldMapper"/>
</EntityMapping>
```

2. Also add the following to file `ab-to-cc-data-mapping.xml` in the `ABContact` section:

```
<FieldMapping source="Tags" mapperClassName="gw.api.util.mapping.NullFieldMapper"/>
```

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.

First time you click on the arrow of the typekey input, the drop-down menu will not open (PL-10134)

Issue: The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting `InputHelpTextOnFocus` to `false` in the `config.xml` file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

Studio Rules do not use correct capitalization for root object's name (PL-10740)

Issue: Rule set root objects are not named with first letter lower-cased.

Workaround: Rules engine will issue warnings when correct case for objects is not being used.

Geocode plugin calls 'isSufficientlyCompleteToGeocode' method twice for each address (PL-11578)

Issue: The Geocode plugin calls the `isSufficientlyCompleteToGeocode` method twice for each address.

Workaround: Guidewire is aware of this issue.

Countries configured in 'zone-config.xml' still generate a warning during regen-dictionary even when zone data is loaded for all of these countries (PL-11947)

Issue: Countries configured in `zone-config.xml` still generate a warning during regen-dictionary even when zone data is loaded for all of these countries.

Workaround: Warning message is created in error and can safely be ignored.

User interface cannot handle starting multiple instances of a batch process (PL-12372)

Issue: The user interface cannot handle multiple instances of a batch process.

Workaround: If you need to execute multiple instances of a batch process, then you need to start each from the command line. Also, you need to ensure that the `BatchProcess.isExclusive` method returns `false` to allow the application to run multiple instances simultaneously.

Type system refresh after PCF page title change does not update corresponding menu label (PL-13057)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

Length limitation on entity localization table names (PL-13360)

Issue: There is a length limitation on entity localization table names.

Workaround: Ensure that localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

'US-Locations.txt' file with the U.S. geodata from GreatData has special characters that cause validation problems with United States Postal Service (USPS) data (PL-13384)

Issue: The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can process the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

Email with file attachment with unicode filename not correctly handed over to the mail server (PL-13582)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

The GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

'JavaToolkit.gs' has incorrectly hard coded memory which results in failed 'regen-java-api' Ant task (PL-13663)

Issue: The JavaToolkit.gs has hard coded memory which can result in failed regen-java-api Ant tasks.

Workaround: Increase the size of the maximum heap setting on line 161 of JavaToolkit.gs in the Ant module. The default value is 512.

Cannot make a delegate field a localized column (PL-13761)

Issue: You cannot make a delegate field a localized column.

Workaround: Move the column that you want to localize off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define an enhancement property on the delegate, that *delegates* to the appropriate column, depending on the implementing entity.

Cannot print second-level ListView (PL-14640)

Issue: Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all of the items.

Workaround: Remove the lower level list view out of the hierarchical structure and include it at the top level.

RowSet and RowIterator configured in the same ListView do not work correctly. (PL-14946)

Issue: Configuring a RowSet and RowIterator in the same ListView does not provide the right behavior for editability.

Workaround: Separate the RowSet and RowIterator that exist in the same ListView into separate components or into multiple ListView widgets.

Masked input fields do not correctly handle ENTER keypress (PL-14955)

Issue: Input fields that have a field validator mask do not correctly handle the ENTER keypress. After you press ENTER, the field no longer displays the input mask and it is not possible to enter any further input.

Workaround: Cancel the page and re-enter the information.

ClaimCenter does not recognize modified 'Calendar.js' file (PL-14997)

Issue: ClaimCenter does not correctly recognize changes made to Calendar.js through Studio. You access this file in the following Studio location:

Resources → Web Resources → resources → javascript → global

Workaround: Instead, modify `global.js` directly. You access this file in the following installation file location:

`ClaimCenter/webapps/cc/resources/javascript`

WARNING This is one of the very few occasions in which you modify an application file outside of the application configuration folder. Ensure that you modify `global.js` only. Otherwise, you can invalidate your installation.

chapter 40

Guidewire ClaimCenter 6.0.7 Release Notes

Release 6.0.7

Copyright © 2001-2013 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Live, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

This material is Guidewire proprietary and confidential. The contents of this material, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

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

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

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
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

Version Number

This release of Guidewire ClaimCenter is 6.0.7.

For Standard Reporting, this release supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *ClaimCenter Reporting Guide* if you plan to use the integrated reporting feature
- The *ClaimCenter Contact Management Guide* to install ContactCenter
- Prior ClaimCenter release notes for any versions that have been skipped

The following issues pertain to installing this release.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-72C-ERX-0000BF00100001F-F2B025280327

If you want to use InetSoft and the ClaimCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *ClaimCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire ClaimCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Support

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

- At the Guidewire Resource Center – <http://guidewire.custhelp.com>
- By email – support@guidewire.com
- By phone – +1-650-356-4955

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Rules Changes
- Changes in This Release Provided in Upgrade Diff Report
- Improvements and General Issues

- Known Issues and Limitations

Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

ClaimCenter release 6.0.6 to 6.0.7

- 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/pl` directory, [click here](#).

Rules Changes

ClaimCenter release 6.0.6 to 6.0.7

- 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.

Updating Localization modules

Note: This topic is applicable only if you are using the Guidewire Localization modules (GB, AU/NZ, JP).

Any display keys that Guidewire changed or added to this maintenance release have not been updated in your localization module. On server startup, ClaimCenter logs warnings for display keys that are missing or updated with different a number of parameters in the localization module of the default application locale. Use the upgrade diff report to find edited or removed keys and update them in your configuration. For more information, see the *ClaimCenter Upgrade Guide* topics named “Translating New Display Properties and Typecodes”. These topics describe how to export, redefine, and import localized values for added and changed display properties and typecodes when you upgrade from various releases of ClaimCenter.

Improvements and General Issues

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

ID	Description
Activities	
PL-19664	Fixed an issue that caused the Activity Escalation work queue writer to create duplicate work items.
Archiving	
PL-17430	Added a configuration upgrade step to automatically remove usages of the <code>AdminTable</code> delegate from your extension entity definitions during a configuration upgrade. The <code>AdminTable</code> delegate became obsolete and was removed during implementation of the File Based Archiving feature in ClaimCenter 6.0.5.
PL-18269	Added a Download button to the (Server Tools) Archive Info page to more easily enable the collection and sending of archiving error information.
PL-18997	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> in a production environment if you make changes to the data model and restart the application server.
Assignment	
PL-18195	Fixed an issue with <code>assignGroupByRoundRobin</code> that reset the round-robin index each time the Group type changed.
Batch Processes	

ID	Description
PL-14787	Guidewire now logs the <code>ConcurrentDataChangeExceptions</code> generated by distributed workers at different levels depending on context. If the <code>ConcurrentDataChangeException</code> occurs on processing the work items, ClaimCenter logs the error only if the number of attempts exceeds the configured value of the <code>WorkItemRetryLimit</code> . Otherwise, ClaimCenter logs the debug message instead.
PL-18020	Converted an example batch process (<code>PurgeFailedWorkItems.gs</code>) to be usable in a production environment.
Claim Metrics	
CLM-17777	<p>Previously, you could not configure the claim metrics update process to run only on certain claims. This fix provides an entry point into the process.</p> <p>The interface delegate <code>ClaimUpdateMetricsMethod.java</code> was added to the <code>Claim</code> entity and implemented in Gosu in <code>gw.claim.metric.ClaimUpdateMetricsMethodImpl.gs</code>. In the base configuration, the delegate calls through to <code>ClaimHealthUpdater.updateClaim</code>, preserving the original base configuration behavior. To configure this feature, you can add your own implementation of <code>ClaimUpdateMetricsMethod</code> or alter the existing <code>ClaimUpdateMetricsMethodImpl</code> class.</p>
Clustering	
PL-19009	Guidewire has upgraded the JGroups version to 2.12.1.
Configuration Upgrade	
PL-12303	Fixed an issue in which the upgrade tool did not handle rules upgrade properly.
Contact Domain	
PL-18244	Fixed issue with <code>IgnoreProperty</code> used in <code>contact-sync-config.xml</code> that caused it to take affect on all subtypes of <code>Contact</code> , rather than only those specified. This functionality now works properly and allows fields to be ignored on <code>contact sync</code> only for the specified subtypes.
PL-19532	The <code>CC_AUTOSYNWORKITEM</code> table has a foreign key relationship from the <code>MAXCONTACTID</code> and <code>MINCONTACTID</code> columns to <code>CC_CONTACT (ID)</code> . This constraint has been removed because the maximum and minimum <code>Contact</code> IDs do not point to particular contacts, but rather indicate a range.
PL-19533	Added configuration parameter <code>ContactAutoSyncWorkItemChunkSize</code> . Use this configuration parameter to set the chunk size for <code>ContactAutoSync</code> work items.
Database Instrumentation	
PL-19629	Fixed an issue in which Oracle AWR (Automatic Workload Repository) download used a hard-coded MKTG user when looking for index on an Oracle materialized view.
Database Support - Oracle	
PL-18448	<p>Added a new configuration parameter—<code>QueryRewriteForClaimSearch</code>—that enables options for query rewrite on Oracle. There are options to rewrite queries using materialized view based on cost or to force the rewrite. The four possible values are as follows:</p> <ul style="list-style-type: none"> <code><param name="QueryRewriteForClaimSearch" value="COST/NOSTALE"/></code> <code><param name="QueryRewriteForClaimSearch" value="COST/STALE"/></code> <code><param name="QueryRewriteForClaimSearch" value="FORCE/NOSTALE"/></code> <code><param name="QueryRewriteForClaimSearch" value="FORCE/STALE"/></code> <p>If you provide an invalid value, the server ignores it.</p>
PL-19603	Fixed an issue that caused Oracle AWR (Automatic Workload Repository) download to not work correctly if the locale of the user running the download used a comma for a decimal sign.
Database Upgrade	
PL-18372	Fixed an issue that caused upgrade to fail if you modified <code>GwNormalize</code> in <code>collations.xml</code> , and it did not mimic the semantics of the <code>LOWER</code> database function.
Document Management	
PL-18376	Guidewire Document Assistant no longer adds extra section breaks to generated Microsoft Word documents.
PL-18431	Improved performance for retrieval of extremely large documents through the ActiveX Guidewire Document Assistant.
PL-18608	Microsoft Word field forms ("text fill-in field") are now restored correctly when generating a document from an Microsoft Word Template.

ID	Description
PL-18699	New versions of the Guidewire Document Assistant ActiveX control will be downloaded to the client browser: <ul style="list-style-type: none"> • 2.0.48 for the standard control • 2.1.48 for the whitelist version of the control
PL-19781	By default, documents are saved in the file system by using document IDs as file names. To change the default behavior to use document names as file names, add the <code>UseDocumentNameAsFileName</code> parameter to the <code>config-override.xml</code> file and set it to <code>true</code> . For example: <pre><param name="UseDocumentNameAsFileName" value="true"/></pre>
PL-19956	Guidewire template documents that use the Microsoft Word formats could not be generated if Microsoft Word was already open. The Guidewire Document Assistant ActiveX controls change versions from 2.0.48/2.1.49 (whitelist) to 2.0.51/2.1.51. If you manually deploy the control(s), you must deploy the new versions.
PL-20598	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.
PL-20642	ClaimCenter now displays a meaningful error message if an error occurs during document edit.
Documentation	
PL-17987	The extraneous <code>Bundle</code> property on the <code>Bundle</code> object (<code>Bundle.Bundle</code>) has been removed.
Financials - Bulk Invoices	
CLM-16674	Fixed an issue in which a bulk invoice approval or rejection history event shows an incorrect amount.
CLM-18910	Fixed an issue in which attempting to get the associated payment of a voided or stopped bulk invoice's invoice item would throw an exception.
CLM-19166	Fixed an issue in which bulk invoices can get stuck during submission processing, or advance even though one or more invoice items have not finished processing. <p>While this issue with the Bulk Invoice Submission batch process has been fixed, it is still possible for a bulk invoice to get stuck. If a bulk invoice unexpectedly stays in Pending Item Validation status for a long time, an administrator should run the Bulk Invoice Submission batch process again. This condition might indicate a misconfiguration, such as more than one thread configured for the <code>BulkInvoiceSubmission</code> work queue. This work queue must be configured only with one thread.</p>
Financials - Checks	
CLM-18705	Exposed the method <code>Check.prepareForTransfer</code> to make it possible to transfer a check in rules.
Financials - Checks, Payments, Deductibles	
CLM-19709	Introduced a new configuration parameter for <code>config.xml</code> , <code>DeleteCompletedApprovalActivitiesOnEdit</code> , which defaults to <code>true</code> . 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 <code>false</code> , then when resubmitting a check or bulk invoice or when invalidating a bulk invoice, any existing completed approval activities are preserved. Open approval activities will be deleted regardless of the setting.
Geocoding/Proximity Search	
PL-17650	Guidewire currently provides an implementation of the <code>GeocodePlugin</code> that connects to the Microsoft Bing Maps Geocode Service. You must use Java 1.6 with the Bing Maps implementation of this plugin in ClaimCenter 6.0.7.
PL-18178	The <code>MapPoint</code> implementation is deprecated by Guidewire with this release. Microsoft announced plans to retire the <code>MapPoint</code> web service, effective November 18, 2011. If you currently use geocoding features and the <code>MapPoint</code> plugin, you must migrate from <code>MapPoint</code> to Bing Maps.
Gosu	
PL-18317	Guidewire now provides a preload mechanism to support pre-compilation of Gosu classes, as well as other primary classes. This is meant to improve application performance by preloading some of the necessary application types. To support this, Guidewire provides a <code>preload.txt</code> file (in Other Resources) in which you can configure this behavior. See "Preloading Gosu Classes" in the Configuration Guide for more details.

ID	Description
PL-18329	<p>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 when running code in the Gosu Tester:</p> <p>The property [or method], "Xxxx", is not visible under the parser's visibility constraints.</p>
Integration	
PL-19760	<p>Guidewire has added a new configuration parameter, <code>AllowActiveXAutoInstall</code>. This parameter controls whether ClaimCenter automatically installs the Guidewire Document Assistant ActiveX control and the supporting JScript files used by the control.</p> <p>If <code>AllowActiveXAutoInstall</code> is set to <code>false</code> and <code>AllowActiveX</code> is <code>true</code>, then ClaimCenter does not install the control, but will use the control if it has been manually installed. Contact Guidewire Support for information about manually installing the control.</p> <p>In this fix, Guidewire has also:</p> <ul style="list-style-type: none"> Added the ability to customize the location of JScript support files used by the control. See "Specifying Location for Guidewire Document Assistant Scripts" in the <i>System Administration Guide</i>. Updated the version number of the control from 2.0.48/2.1.48 (whitelist) to 2.0.53/2.1.53 (whitelist).
Localization	
PL-13411	ClaimCenter now adjusts for daylight savings time in Cuba properly.
PL-13914	ClaimCenter no longer uses daylight savings time in Russia.
Logging	
PL-19578	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.
Messaging	
PL-17120	Improved the logic behind the Event Message administration page to make it more resilient if a message destination becomes unresponsive.
PL-17291	<p>ClaimCenter now provides information on Message and Message History that includes:</p> <ul style="list-style-type: none"> The time it takes to run the <code>beforeSend</code> transformation (including the lock time) The time it takes to run the <code>send</code> method (including the lock time) The time it takes to run the <code>afterSend</code> method The time it takes to acknowledge the message
PL-18293	ClaimCenter now generates the correct message send order, even if a bundle commit contains multiple event root entities.
Miscellaneous	
CLM-18201	The populator for <code>ClaimAccess</code> staging tables was changed to use the <code>PublicIDPrefix</code> configuration parameter in <code>config.xml</code> . Previously this populator was hard-coded to use the default <code>cc</code> prefix.
CLM-18486	<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>This problem is fixed for ClaimCenter 6.0.7, and the vacation tab functionality should be back to the way it was in 6.0.5. Performance might be marginally better now that the duplicate predicates have been removed.</p> <p>If you are upgrading from 6.0.6 and you used the Guidewire 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, and the new functions are exact replacements for the functions provided by the workaround enhancements. Therefore, you will not need to make further changes to your PCF files as a result of this change.</p>
CLM-19676	Fixed the <code>Claim</code> property on the <code>SubroAdverseParty</code> entity, which would throw a null pointer exception (NPE) when the entity was not hooked up to its <code>SubrogationSummary</code> . The property now returns <code>null</code> in this case. The previous behavior caused a problem when restoring an archived claim if the <code>SubroInstallmentAmount</code> field on the <code>SubroPaymentSchedule</code> entity was populated. The same problem was also fixed in the properties <code>StatuteLimitationsLine.Claim</code> and <code>MatterView.ClaimCurrency</code> .

ID	Description
Other - Database	
PL-18618	Prior to this change, configuration parameter <code>DisableIndexFastFullScanForProximitySearch</code> controlled the disabling of Oracle access paths such as index fast full scan and hash join while performing a proximity search using Oracle database. With this change, configuration parameter <code>DisableHashJoinForProximitySearch</code> controls the disabling of hash join during proximity search.
Other - PL Services	
PL-11582	Added an error log entry stating that Jetty is not a supported application server.
Profiling	
PL-14608	The (Server Tools) Guidewire Profiler now supports a csv version of a stack to a file.
PL-17534	Fixed an issue with the (Server Tools) Web Profiler in which the Group Frames result page did not show detail information. The page now shows the detail information.
PL-17636	Modified the (Server Tools) Guidewire Profiler page: <ul style="list-style-type: none"> • Moved the Diff DBMS Instrumentation Counters column to after the Extended Query Tracing column. • Provided the ability to click x in individual columns for a work queue entry point to turn on that Individual Stack. • Provided the ability to check-mark individual columns for a work queue entry point to turn off that Individual Stack.
Server Lifecycle	
PL-18089	ClaimCenter now ensures that the application server's type system is not refreshed in production mode.
Studio IDE - Debugger	
PL-18035	Fixed an issue that caused stepping through Gosu code in the Studio debugger to be much slower than it should have been. The fix included adding diagnostic timing that you can enable by setting logger category Configuration to the DEBUG level. The server console then displays the debugger run/stepping overhead. <p>To temporarily disable the optimization, yet still see timing data, add the following to the server launch command:</p> <pre>-Dstudio_disable_debug_opts=true</pre>
Templates	
PL-18294	If you have used the <code>\${x}</code> substitution in document templates, then you need to revert those uses back to the <code><%= ... %></code> syntax.
Utilities	
PL-17761	Corrected the numerical count of active sessions for active users. Previously, active sessions were being timed out for the count, even though users were still active.
Web - UI/Runtime	
PL-18090	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.
CLM-18321	Fixed an issue when using Internet Explorer 9 and clicking a drop-down list did not render the list properly. Explorer 9 must <i>not</i> be in Compatibility Mode for this fix to work.
XML Element (and XSD types)	
PL-17898	Modified Studio to correctly parse complex types within circularly referencing XSDs.

Known Issues and Limitations

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

- ClaimCenter Known Issues
- Platform/Studio Known Issues

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

Integration between ClaimCenter 6.0.7 and PolicyCenter 7.0.2 does not work (PC-15581)

Issue – The PolicyCenter 7.0.2 release does not contain a required RPC version of the ClaimToPolicySystemNotificationAPI web service that ClaimCenter 6.0.7 needs for integration with PolicyCenter 7.0.2.

Workaround – Contact Guidewire Support for the proper version of this API.

Exception in the Bulk Invoice screen after restoring and reopening an archived claim (CLM-174)

Issue – If you archive a claim, restore it within 60 minutes of archiving it, and then immediately attempt to view a bulk invoice involving that claim, the system may show an exception message. This is because it takes time for the global cache to be updated after the claim is archived. However, this does not cause any data loss or corruption.

Workaround – You must wait 60 minutes after archiving a claim before restoring it. The length of time to wait is determined by the GlobalCacheStaleTimeMinutes configuration parameter. If an error does occur, log out (and then you may log back in). Do not attempt to view the bulk invoice for 60 minutes. Guidewire may not fix this issue, since restoring archived claims so soon after they were archived is considered a very rare occurrence. No data loss or corruption occurs, and the fix may cause performance problems with archiving.

Link issues between a new contact from the New Check wizard and ContactCenter (CLM-12172)

Issue – If you try to link a contact newly created from the New Check wizard to ContactCenter, the contact details can be overridden if the contact already exists in ContactCenter. This occurs if there is a match between the two integrated systems. Rather than creating a new contact, the existing matching contact information in ContactCenter overwrites the contact's information in ClaimCenter.

Workaround – Create the contact first in ClaimCenter (not in the New Check wizard), then if you want, link it to ContactCenter. Finally, use the New Check wizard to create a check for that contact.

Tax ID fields are not masked in claim snapshots that were created in earlier versions of ClaimCenter (CLM-12879)

Issue – Tax ID fields in the claim snapshot are not masked in the user interface. This happens because the claim snapshot is using an earlier version of the ClaimSnapshotContactxxx.pcf file which does not have the field encryption on the claim snapshot. (xxx represents various versions of the pcf file.)

Workaround – You must manually add a mask to the encrypted fields in earlier versions (such as 5.0.0) on the ClaimSnapshotContactxxx.pcf file.

Unable to generate a Claim Metrics report through the InetSoft Report Portal (CLM-13141)

Issue – The InetSoft onLoad script is designed to be run from within ClaimCenter and it should pass the report the locale. If the report is run outside of ClaimCenter, the locale needs to be set within InetSoft.

Workaround – Run reports through ClaimCenter.

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.

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 `RangedInput` widget on `ClaimSearchDV` and set `labelAbove="true"`.

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`):

1. Delete the following entry:

```
log4j.additivity.PluginsLog=false
```

2. Add the following entry in its place:

```
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.

For example, the following statements need to have `Log` removed from the name, as shown after the arrow:

- `log4j.additivity.APILog=false` → `log4j.additivity.API=false`
- `log4j.additivity.DatabaseLog=false` → `log4j.additivity.Database=false`
- `log4j.additivity.MessagingLog=false` → `log4j.additivity.Messaging=false`
- `log4j.additivity.ClaimNumGenLog=false` → `log4j.additivity.ClaimNumGen=false`
- `log4j.additivity.PolicySearchLog=false` → `log4j.additivity.PolicySearch=false`
- `log4j.additivity.ContactSearchLog=false` → `log4j.additivity.ContactSearch=false`

Localized document pattern specified in the activity pattern is not passed to the activity (CLM-13670)

Issue – When ClaimCenter is localized to the Japanese language, and you create an activity using the *Gosu Sample Email Sent Record* activity pattern, the system does not select the correct language.

Workaround – Use the document search feature to find the applicable localized document template.

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`.

Workaround – Use destination IDs instead of message sink IDs if you can. If you cannot use message sink IDs, then continue to use the deprecated methods.

Must regenerate database statistics after large batch processing of addresses (CLM-13963)

Issue – After geocoding a large batch of addresses—new addresses greater than 10% of the current addresses in the `ContactCenter` database—you must run database statistics on the `ab_abaddress` table. Running database statistics helps the database identify the distribution of the addresses and thus improves query performance.

Workaround – To generate a file containing the database commands, at a command prompt, navigate to `ContactManager/admin/bin` and run the following command:

```
maintenance_tools -password password -getdbstatisticsstatements | grep -i ab_abaddress > filename
```

In this command, *password* is your administration password and *filename* is the file where you want the output of the command to be saved. The `grep` command is available on UNIX or Linux systems. On Windows systems, you can run `grep` in a UNIX or Linux emulator like Cygwin.

The database administrator must then run the commands contained in *filename*.

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 following `onToggle` attribute to this file:

```
onToggle="InsuredVehicle.Checked = !InsuredVehicle.Checked; Wizard.synchronizeVehicleIncidents(Claim)
```

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 on 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 function is executed on the `MainContacts` 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 `MainContacts` wizard step call the new function, as follows:

```
onExit="checkForDuplicatesNoWarning()"
```

Now, a user can go to the next step. However, the user is still warned of duplicate claims encountered. The **Next** button proceeds to the next step in the wizard and avoids extra queries to the external policy system.

Duplicate descriptions for image/jpg mime types in the configuration file (CLM-14188)

Issue – In the file `modules\p1\config\locale\en_US\display.properties`, there are two entries under MIME types for jpeg file types that are defined with the same description, **JPEG Image**:

- `pjpeg`
- `jpeg`

These definitions cause a problem because the descriptions of both mime types are identical. If you have the **New Document** screen open to add a new image document and you click the **Document Type** drop-down list, you see two entries for **JPEG Image**. If you choose the wrong image type and add the new document, an error occurs.

Workaround – Open Guidewire Studio and do the following:

1. Navigate in the left pane to **configuration** → **Display Keys** → **Mimetype** → **Configuration** → **image_pjpeg**.
2. Change the text defined in **Locale: English (US)** to **Progressive JPEG Image**.
3. Save your changes and restart ClaimCenter.
4. Now in the ClaimCenter **New Document** screen, when you click the **Document Type** drop-down list, you see two choices for these file types, **JPEG Image** and **Progressive JPEG Image**.

No warning that you will overwrite a Service Provider Review or Question Set (CLM-14365)

Issue – If you do not retire the review or question set and import the new review or question set, then there is no warning or mechanism to prevent you from overwriting the existing review or question set.

Workaround – You should create the new review or question set in a new .csv file. Also, ensure that all public IDs of all entities of the new review or question set are unique and different from the existing review or question set. Consider that any entities such as **ReviewType**, **ReviewCategoryQuestionSet**, **QuestionSet**, **Question**, **QuesitonChoice**, and **QuestionFilter** could be overwritten.

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.

Manual exchange rate printing is incorrect (CLM-14903)

Issue – If you print any financials pages in multicurrency mode, the manual exchange rate printout is incorrect. The issue is caused by incorrect printing when using the **format** attribute on a PCF widget.

Workaround – You must correct the **TransactionExchangeRateInputSet.pcf** file in Guidewire Studio. In this PCF file, you change the **Exchange Rate** input widget, which has ID **Transaction_ExchangeRate**, and add a new **TextInput** widget.

To make these changes in Guidewire Studio:

1. Navigate to **configuration** → **Page Configuration (PCF)** → **claim** → **shared** and click **TransactionExchangeRateInputSet** to open this PCF file in the PCF editor.
2. Click the **Exchange Rate** widget with ID **Transaction_ExchangeRate** to open its **Properties** pane.
3. In the **Properties** pane click **Advanced properties** and set the **visible** property to:


```
not (print.PrintUtil.isPrintingPDF()) or print.PrintUtil.isPrintingCSV())
```

 This code hides this version of the **Exchange Rate** field during a print operation.
4. Drag a **TextInput** widget from the **Toolbox** and drop it below the current **Exchange Rate** widget.
5. Click the new **TextInput** widget and set the following properties:
 - **id** – **Transaction_ExchangeRateForPrinting**
 - **label** – **displaykey.Web.Claim.ExchangeRateInputSet.ExchangeRate**
 - **outputConversion** – **return "1 " + VALUE.BaseCurrency.DisplayName + " = " + new java.text.DecimalFormat("#.#####").format(VALUE.Rate) + " " + VALUE.PriceCurrency.DisplayName**
 - **value** – **transaction.TransToClaimExchangeRate**

- `visible` – `print.PrintUtil.isPrintingPDF()` or `print.PrintUtil.isPrintingCSV()`

Note: These changes are one example of how to get printing to perform correctly in multicurrency mode. For example, you might want to use different code for the `outputConversion` property, depending on your configuration of ClaimCenter.

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.

UnsupportedOperationException: MetaIntrinsicType does not support array yet (CLM-15718)

Issue – ClaimCenter 3.x snapshot PCF files generate an exception saying `MetaIntrinsicType` does not support array yet if the snapshot PCF file contains the following cell:

```
Cell id="Lienholders" value="util.Snapshot.renderValue(PolicyVehicle.Vehicle.Lienholders)"
label="displaykey.Web.ClaimSnapshotPolicyVehicles300LV.Lienholders"/>
```

You see this problem if you navigate to the **FNOL Snapshot** → **Policy** → **Vehicle** screen and select a ClaimCenter 3.x snapshot claim that contains this particular cell.

Workaround – In Guidewire Studio, navigate in the **Resources** pane on the left to the following location and delete this particular `Lienholders` cell from any PCF file that contains it.

`configuration` → `Page Configuration (PCF)` → `claim` → `snapshot` → `300`

For example, if you click `ClaimSnapshotPolicyVehicles300LV` and open it in the PCF editor, you see a `Lienholders` cell that has the same property settings as the cell described previously. Right-click the cell and click **Delete**. If you see a message asking if you want to edit the PCF file, click **Yes**. Then save your changes.

ClaimCenter does not prevent concurrent restore of archived claim by second user (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` exception before returning the archived claim document to be restored.

Existence of table `cc_tmpcheckrpt` can cause upgrade to fail (CLM-16668)

Issue – If you ran the `FinancialsCalc` batch process in a pre-5.0.8 version of ClaimCenter, it is possible that it populated a temporary table, `cc_tmpcheckrpt`. The 6.0.7 upgrade process does not update this table successfully, which causes the upgrade to fail. The `FinancialsCalc` batch process does not run automatically in the base configuration. You must trigger it manually through the maintenance tools. For example:

```
maintenance_tool financialscalculations
```

Workaround – Delete table `cc_tmpcheckrpt` before upgrading. The batch process uses the contents of this table only while the process is actively running. However, the batch process does not remove the contents of the table after it completes. You can safely delete this table.

Unable to archive a claim after performing policy refresh on it (CLM-16708)

Issue – If you attempt to archive a claim after performing a policy refresh on it, the archive attempt fails. The policy refresh process does not detect and sever all connections between the old policy graph and the claim, and these connections prevent the claim from being archived. If you want to implement both claim archiving and policy refresh, you need to sever these connections after policy refresh has run.

Workaround – In Guidewire Studio, refer to the class `RetiredPolicyGraphDisconnecterExample` for an example of the type of code to execute after policy refresh has run to sever these connections. You can also refer to Knowledge Base article #1555 for more information.

Statistics batch process queries do not scale on large Oracle databases (CLM-16830)

Issue – Guidewire has determined that some of the queries used by the dashboard statistics batch process (`DashboardStatistics`) do not scale well on large Oracle databases. The dashboard statistics batch process can take up to 14 hours on a 5 million-claim multi-byte database. Taking this much time is problematic, even though the statistics batch process runs just once a night and other tasks appear to complete normally while it is running.

Workaround – It is possible to improve performance by doing the following:

1. Alter an index on `Transaction` to include more columns, which reduces table accesses by the dashboard statistics queries.
2. For Oracle, use Oracle Stored Outlines to force a better plan for some of the dashboard statistics queries.

To review details of Oracle performance enhancements:

1. Navigate to the following link on the Resource Portal:
<http://guidewire.custhelp.com/app/resources/infrastructure/documents>
2. Select the following white paper from the list under **Guidewire Infrastructure Whitepapers**:
ClaimCenter 6.0 Using Oracle.pdf

Method `adjustDocumentName` can cause an infinite loop (CLM-17262)

Issue – Previously, `ClaimCenter` used `DocumentProduction.adjustDocumentName` to append (1), (2),... (n) to any documents that have the same name as a document that already exists in `ClaimCenter`. This method works properly if the document name is the key used to uniquely identify the document. However, if the document ID is used instead as the unique identifier, the system goes into an infinite loop.

To help remedy this issue, Guidewire has moved method `adjustDocumentName` from `DocumentProduction` to `DocumentEnhancement` to make it possible for you to remove or edit this method as needed.

Workaround – Modify `gw.document.DocumentEnhancement` and change the logic that performs the document rename. The logic to change is in a `while` loop in the method `adjustDocumentNameIfDuplicate`.

Integration between ClaimCenter 6.0.7 and ContactManager not working (CLM-17933)

Issue – `ClaimCenter` does not properly handle the integration between `ClaimCenter 6.0.7` and `ContactManager 7.0.0`.

Workaround – Add the following entries to the `ab-to-cc-data-mapping.xml` file in `ClaimCenter 6.0.7`:

1. Add the following:

```
<EntityMapping source="ABContactTag" target="ContactTag">
  <FieldMapping source="ABContact" mapperClassName="gw.api.util.mapping.NullFieldMapper"/>
</EntityMapping>
```
2. Also add the following to the `ABContact` section:

```
<FieldMapping source="Tags" mapperClassName="gw.api.util.mapping.NullFieldMapper"/>
```

Bulk invoice stuck in Pending Bulk Invoice Item Validation status (CLM-19715)

Issue – A bulk invoice that has been submitted is stuck in *Pending Bulk Invoice Item Validation* status and all workers have finished,

Workaround – Run the Bulk Invoice Submission batch process again. Additionally, this condition might indicate a misconfiguration, such as more than one thread configured for the `BulkInvoiceSubmission` work queue. This work queue must be configured with only one thread.

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.

First time you click 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 does not interfere with opening a drop-down menu.

ListDetailPanel throws exception (PL-10316)

Issue – It is possible for ClaimCenter to throw an exception if the user cancels out of a `ListDetailPanel` widget and if `StartInEditMode` is also set to `true`.

Workaround – Set `StartInEditMode` to `false` for the screen that contains the `ListDetailPanel`. As a consequence, the user must click **Edit** to modify that screen.

Studio Rules do not use correct capitalization for root object's name (PL-10740)

Issue – Rule set root objects are not named with first letter lower-cased.

Workaround – Rules engine will issue warnings when correct case for objects is not being used.

Geocode plugin calls `isSufficientlyCompleteToGeocode` method twice for each address (PL-11578)

Issue – The Geocode plugin calls the `isSufficientlyCompleteToGeocode` method twice for each address.

Workaround – Guidewire is aware of this issue.

User interface cannot handle starting multiple instances of a batch process (PL-12372)

Issue – The user interface cannot handle multiple instances of a batch process.

Workaround – If you need to execute multiple instances of a batch process, then you need to start each from the command line. Also, you need to ensure that the `BatchProcess.isExclusive` method returns `false` to allow the application to run multiple instances simultaneously.

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 corresponding menu label.

Workaround – After changing a page title and saving it in Studio, you must 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 localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

US-Locations.txt file with the U.S. geodata from GreatData has special characters that cause validation problems (PL-13384)

Issue – The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround – The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can process the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue – When GX models reference virtual fields and enhancements, they throw null pointers when the references are null.

Workaround – When you implement a GX model, ensure that you include null checks and error handling for virtual fields or enhancements that might be null, to avoid null pointer exceptions.

Email with file attachment with unicode filename not correctly handed over to the mail server (PL-13582)

Issue – Sending email with a file attachment that has a unicode filename is not sent to the mail server correctly.

Workaround – Use Latin characters for file names on attached files.

The GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue – The GX model generated XSD cannot be parsed by JAXB.

Workaround – Add JAXB annotation elements to the XSD to specify the necessary metadata, such as class names, so that JAXB can generate the Java class files. Contact Guidewire Support for a sample XSD file that is annotated in this way.

JavaToolkit.gs has incorrectly hard coded memory which results in failed regen-java-api Ant task (PL-13663)

Issue – The `JavaToolkit.gs` has hard coded memory, which can result in failed `regen-java-api` Ant tasks.

Workaround – Increase the size of the maximum heap setting on line 161 of `JavaToolkit.gs` in the Ant module. The default value is `:MemoryMaximumSize = 512`. Set this value to at least 768.

Cannot make a delegate field be a localized column (PL-13761)

Issue – You cannot make a delegate field be a localized column.

Workaround – Move the column that you want to localize off the delegate and onto each of the implementing entities. To make the column appear as though it exists on the delegate, define an enhancement property on the delegate that *delegates* to the appropriate column, depending on the implementing entity.

Cannot print second-level ListView (PL-14640)

Issue – Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all of the items.

Workaround – Remove the lower level list view from the hierarchical structure and include it at the top level.

RowSet and RowIterator configured in the same list view do not work correctly. (PL-14946)

Issue – Configuring a RowSet and RowIterator in the same ListView widget does not provide the right behavior for editability.

Workaround – Separate the RowSet and RowIterator that are in the same ListView into separate components or into multiple ListView widgets.

Masked input fields do not correctly handle ENTER keypress (PL-14955)

Issue – Input fields that have a field validator mask do not correctly handle the ENTER keypress. After you press ENTER, the field no longer displays the input mask, and it is not possible to enter any further input.

Workaround – Cancel the page and re-enter the information.

ClaimCenter does not recognize modified Calendar.js file (PL-14997)

Issue – ClaimCenter does not correctly recognize changes made to the Calendar.js file in Guidewire Studio. You access this file in the following Studio location:

configuration → Web Resources → resources → javascript → global

Workaround – Modify the global.js file directly. This file has a Calendar.js section that has the same code as the Calendar.js file.

The global.js file is located in the following directory:

ClaimCenter/webapps/cc/resources/javascript

WARNING This file is one of the very few that you can modify directly, outside the application configuration folder. Ensure that you modify global.js only. Otherwise, you can invalidate your installation.

Studio debugger cannot debug gw.sampledata classes (PL-15357)

Issue – Attempting to debug a test that loads or uses base application sample data causes an unimplemented bytecode error in the JVM. This means that you cannot use the Studio debugger to debug gw.sampledata classes.

Workaround – Guidewire is aware of this issue.

Cannot print second-level list view (PL-15548)

Issue – Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all the items.

Workaround – Remove the lower level list view from the hierarchical structure and include it at the top level.

Errors with non-negative currency amounts passed between applications (PL-20112)

Issue – Integration code between ClaimCenter and ContactCenter throws a `ClassNotFoundException` for `entity.CurrencyAmount` whenever you create a non-negative currency amount column in both applications.

Workaround – Define the columns as type `varchar` instead of type `nonnegativecurrencyamount`.

...in metadata

The following sample metadata XML code defines a non-negative currency amount column.

```
<column
  desc="Agreed rate amount for the Service Provider"
  name="CGU_XXX"
  type="nonnegativecurrencyamount"/>
```

Instead, use the following sample metadata XML code, which defines a `varchar` column to use in place of a `nonnegativecurrencyamount` column.

```
<column
  desc="Agreed rate amount for the Service Provider"
  name="CGU_XXX"
  type="varchar">
  <columnParam
    name="size"
    value="10"/>
</column>
```

Depending on the currency you want to use, you might need to adjust the `size` attribute of the `columnParam` element the metadata definition.

...in page configuration

1. Use a text input field.
2. For the `RequestValidationExpression` parameter under **Advanced Properties**, specify the following formula.

```
(VALUE.matches("\\d*\\.\\d\\d") ? null : "Invalid value - must be <digit>*.<digit><digit>")
```

The preceding expression ensures that the entered value contains zero or more digits, followed by a decimal point, followed by two digits. If the expression evaluates to `true`, the application validates the entered value successfully. If the expression evaluates to `false`, the application rejects the entered value and displays the message specified after the colon (:).

Depending on the currency you want to use, you might need to adjust the `matches` mask and the error message.

