

Guidewire PolicyCenter®

PolicyCenter Application Guide

RELEASE 8.0.0

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 Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, 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>).

Product Name: Guidewire PolicyCenter

Product Release: 8.0.0

Document Name: PolicyCenter Application Guide

Document Revision: 17-September-2013

Contents

About This Document.....	25
Intended Audience	25
Assumed Knowledge	25
Related Documents.....	25
Conventions in This Document	27
Support	27

Part I Introduction

1 Introduction to PolicyCenter	31
Product Model.....	32
Lines of Business in the Base Application	32
The Policy Lifecycle.....	33
PolicyCenter Integration Points	34
Integration with Other Guidewire Applications	35
PolicyCenter Users	36

Part II PolicyCenter User Interface

2 Navigating PolicyCenter.....	39
Logging into PolicyCenter	39
Setting Preferences	40
Selecting International Settings in PolicyCenter	40
Common Areas in the PolicyCenter User Interface	42
PolicyCenter Tabs.....	43
Desktop Tab	43
Account Tab	46
Policy Tab	47
Contact Tab.....	47
Reinsurance Tab	48
Search Tab.....	48
Team Tab	49
Administration Tab	49
3 QuickJump.....	51
QuickJump Overview	51
Jump to a Screen or Entity	51
QuickJump Behavior in Wizards	52
Using QuickJump	52
Chaining QuickJump Destinations Together	52
Configuring QuickJump	52
QuickJump References	53

4 Basic Search	55
Basic Search Overview.....	55
Indexing Free-text Search Data.....	56
Query and Filter Search Fields	56
Exact or Inexact Search and Ranking.....	56
Search Results for Fields that Match Two or More Similar Pieces of Information	56
Basic Policy Search Overview	57
Basic Search User Interface	57
Policy Basic Search Screen.....	57
Working with Basic Search	60
Basic Policy Search Examples	60
Search Index Updates Automatically.....	60
5 Advanced Search	63
Advanced Search Overview	63
Minimum Search Requirements	64
Advanced Search for Policies	64
Search Contacts.....	65
Working with the Advanced Search Tab	65
6 Saving Your Work.....	67
Unsaved Work Menu	67
Saving Your Work in PolicyCenter Wizards	68
Popup Window Behavior	68
7 Localizing PolicyCenter	69
Localizing Documents, Email, and Notes	69
Localizing Activities.....	69
Localizing Workflows	70

Part III

PolicyCenter Policy Transactions

8 Policy Transactions	73
Overview of Policy Transactions	73
Submission	74
Issuance.....	74
Renewal.....	74
Cancellation	74
Policy Change	75
Reinstatement	75
Rewrite	75
Rewrite New Account.....	75
Audit	75
Key Features of Policy Transactions	76
Preemption	77
Out-of-sequence Changes and Policy Transactions	77
Configuring Policy Transactions	78
9 Submission Policy Transaction.....	79
Submission Overview.....	80
Submission General Steps	81

Submission Key Features	82
Understanding Name Clearance and Risk Reservation	82
The Difference Between Quick Quote and Full Application	82
Answering Pre-qualification Questions	83
Selecting an Underwriting Company	83
Reasons for Underwriter Review in Submissions.....	84
Creating Multiple Versions	84
Closing a Submission	84
Differences Between Binding a Policy and Issuing a Policy	85
Expiring Submissions	85
Copying Submission Information	85
The Submission Manager	86
Working with Submissions	86
Places to Create a Submission	86
Creating a Submission	87
Copying a Submission	88
10 Issuance Policy Transaction	89
Issuance Overview	89
How Issuance Works.....	90
Issuance General Steps	90
Working with Issuances	91
Issuing a Policy.....	91
11 Renewal Policy Transaction.....	93
Renewal Overview	93
Starting Renewals Manually in the User Interface	94
Starting Renewals by Using the Policy Renewal Web Service	94
Starting Renewals by Using a Batch Process	94
Renewal Flows	95
Renewals Create New Policy Periods.....	95
Renewal Restrictions.....	96
Renewal Outcomes	96
Renewal General Steps	96
Renewal Key Features	98
Understanding Pre-renewal Directions.....	98
Understanding Referral Reasons.....	99
Understanding Underwriting Issues	99
Understanding Preemption	100
Creating Multiple Versions.....	100
Working with Pre-renewal Directions	101
Viewing an Existing Pre-renewal Direction	101
Creating a Pre-renewal Direction	101
Deleting a Pre-renewal Direction	102
Working with Underwriting Issues and Referral Reasons	103
Working with Renewals	103
Creating a Renewal from a Batch Process	103
Creating a Manual Renewal	104
Viewing Your Renewals.....	104

12 Cancellation Policy Transaction	107
Cancellation Overview	108
Scheduling Versus Cancel Now	108
Default Cancellation Effective Date	108
Changing a Cancellation	109
Rescinded Versus Withdrawn Cancellations	110
Cancellation General Steps	111
Working with Cancellations	111
Canceling a Policy	112
Rescinding a Cancellation	112
Changing the Cancellation Effective Date on an Open Cancellation	113
Viewing Cancellations in Policy Transactions	113
13 Policy Change Transaction	115
Policy Change Overview	116
Policy Change General Steps	117
Policy Change Key Features	117
Using the Policy Review Screen to Verify Changes	118
Changing the Producer of Service in a Policy Change	118
Changing the Policy Expiration Date in a Policy Change	118
Editing the Effective Date of a Policy Change	118
Handling Out-of-sequence Policy Transactions in a Policy Change	122
Using Preemption in a Policy Change	122
Working with Policy Changes	122
Creating a Policy Change	123
Editing the Policy Change Effective Date	123
14 Reinstatement Policy Transactions	125
Reinstatement Overview	125
Reinstatement General Steps	126
Reinstatement Key Features	126
Working with Restorations	126
Reinstating a Policy	127
15 Rewrite Policy Transactions	129
Rewrite Overview	130
Rewrite General Steps	131
Rewrite Key Features	131
Change Any Policy Information	131
Full-Term Rewrite	131
Mid-Term Rewrite	132
Working with Rewrites	132
Rewriting a Policy	132
Rewriting a Policy with a Lapse in Coverage	132
Approving, Requesting Changes, or Declining a Rewrite	132
Viewing Rewritten Policies	133
16 Rewrite New Account Policy Transaction	135
Rewrite New Account Overview	136
Restrictions to Policy Transactions Started on the Source Policy	136
Restrictions to Policy Transactions Started on the Target Policy	140
Rewrite New Account and the Renewal Batch Process	140
Working with Rewrite New Account	141
Rewriting a Policy to a New Account	141

17 Premium Audit Policy Transaction	143
Final Audit Overview	144
Audit Schedules	144
Where Does the Carrier Send Premium Audit Information?	145
Reversing and Revising Final Audits	145
Adding a Final Audit.	146
Premium Report Overview.....	146
Premium Report Audit Schedules.....	147
Premium Report Trend Analysis.....	147
Adding a Premium Report	147
Audit Roles	148
Premium Audit and Other Policy Transactions	149
Premium Audit General Steps	151
Final Audit General Steps	151
Premium Reports General Steps.....	152
Working with Final Audits	152
Viewing a Final Audit.....	153
Viewing Audit Activities	153
Editing or Waiving a Final Audit	154
Adding a New Audit	154
Starting a Final Audit	154
Entering Audit Data and Completing a Final Audit	155
Revising a Final Audit	155
Working With Premium Reports	156
Viewing a Premium Report.....	156
Viewing Premium Report Activities.....	157
Selecting Premium Reports.....	157
Starting a Premium Report	157
Creating Activities for Overdue Premium Reports.....	157
Entering Premium Report Data.....	158
Editing or Waiving a Premium Report	158
Adding a New Premium Report	158
18 Side-by-side Quoting	159
Side-by-side Quoting Overview	159
Side-by-side Quoting Compared with Multi-Version Quoting.....	159
Side-by-side Availability Overview	160
Base Data Overview	160
Side-by-side Data Overview	161
Side-by-side Quoting Process Flow	162
Side-by-side Quoting in the User Interface	162
Tools Menu Items for Side-by-side Quoting	163
Side-by-side Quoting Screen.	163
Policy Versions Screen	165
Working with Side-by-side Quoting.....	165
Selecting Side-by-side Quoting in a Submission	166
Editing a Version in a Policy Transaction with Side-by-side Quoting.....	166
Binding and Issuing a Side-by-side Submission	167
19 Multi-version Quoting.....	169
Multi-version Quoting and Side-by-side Quoting	169
Working with Multi-version Quoting	169
Creating and Comparing Multi-version Quotes	170
Setting the Maximum Number of Multi-version Quotes	170

Part IV Lines of Business

20 Line of Business Overview	173
Additional Lines of Business	173
Developing a New Line of Business.....	174
21 Commercial Auto	175
Commercial Auto Screens	175
Offerings Screen for Commercial Auto	176
Qualification Screen for Commercial Auto	176
Policy Info Screen for Commercial Auto	176
Locations Screen for Commercial Auto	178
Vehicles Screen for Commercial Auto.....	179
Commercial Auto Line Screen for Commercial Auto	180
State Info Screen for Commercial Auto	181
Drivers Screen for Commercial Auto	182
Covered Vehicles Screen for Commercial Auto.....	182
Modifiers Screen for Commercial Auto	183
Risk Analysis Screen for Commercial Auto.....	183
Policy Review Screen for Commercial Auto	184
Quote Screen for Commercial Auto	184
Forms Screen for Commercial Auto	185
Payment Screen for Commercial Auto	185
Commercial Auto Object Model.....	187
Commercial Auto Object Model Overview	188
Covrances in Commercial Auto	189
Modifiers in Commercial Auto.....	189
Locations in Commercial Auto	189
Drivers in Commercial Auto.....	189
Jurisdictions in Commercial Auto.....	189
22 Businessowners	191
Businessowners Screens	191
Offerings Screen for Businessowners.....	192
Qualification Screen for Businessowners	192
Policy Info Screen for Businessowners.....	192
Businessowners Line Screen.....	194
Locations Screen for Businessowners.....	195
Buildings and Locations Screen for Businessowners	196
Modifiers Screen for Businessowners.....	197
Risk Analysis Screen for Businessowners	197
Policy Review Screen for Businessowners.....	197
Quote Screen for Businessowners	197
Forms Screen for Businessowners	198
Payment Screen for Businessowners	198
Businessowners Object Model	200
Businessowners Object Model Overview.....	201
Covrances in the Businessowners	202
Modifiers in Businessowners	202

23 Commercial Package Policy	203
Commercial Package Screens.....	204
Offerings Screen for Commercial Package.....	204
Qualification Screen for Commercial Package.....	204
Policy Info Screen for Commercial Package	205
Line Selection Screen for Commercial Package.....	206
Locations Screen for Commercial Package	207
Line of Business Screens for Commercial Package	207
Modifiers Screen for Commercial Package	208
Risk Analysis Screen for Commercial Package	208
Quote Screen for Commercial Package	209
Forms Screen for Commercial Package	210
Payment Screen for Commercial Package	210
Commercial Package Object Model	212
Commercial Package Product Model	213
Commercial Package Forms	214
24 Commercial Property	215
Working with Commercial Property.....	215
Adding Locations and Buildings.....	216
Removing Buildings	216
Copying Coverages to Other Buildings	216
Commercial Property Screens	216
Policy Info Screen for Commercial Property	217
Buildings and Locations Screen for Commercial Property.....	218
Blankets Screen for Commercial Property	219
Modifiers Screen for Commercial Property	220
Risk Analysis Screen for Commercial Property	220
Policy Review Screen for Commercial Property	221
Quote Screen for Commercial Property	221
Forms Screen for Commercial Property	222
Payment Screen for Commercial Property	222
Commercial Property Object Model.....	225
Line Entity in Commercial Property.....	226
Location Entity in Commercial Property	226
Building Entity in Commercial Property	226
Coverage Entities in Commercial Property.....	226
Modifier Entity in Commercial Property	227
Blanket Entities in Commercial Property	228
Configuring Commercial Property	229
Configuring Copying of Coverages for Commercial Property.....	229
25 General Liability	231
General Liability Overview	231

General Liability Screens	232
Qualification Screen for General Liability	232
Policy Info Screen for General Liability	232
Locations Screen for General Liability	234
Coverages Screen for General Liability	234
Exposures Screen for General Liability	237
Modifiers Screen for General Liability	238
Risk Analysis Screen for General Liability	238
Policy Review Screen for General Liability	238
Quote Screen for General Liability	239
Forms Screen for General Liability	239
Payment Screen for General Liability	240
General Liability Object Model	242
Line Entity in General Liability	243
Coverage Entity in General Liability	243
Modifier Entity in General Liability	243
26 Inland Marine	245
Inland Marine Overview	246
Inland Marine Screens	246
Policy Info Screen for Inland Marine	246
Coverage Part Selection Screen for Inland Marine	248
Buildings and Locations Screen for Inland Marine	248
Accounts Receivable Screen for Inland Marine	249
Contractors Equipment Screen for Inland Marine	249
Signs Screen for Inland Marine	250
Risk Analysis Screen for Inland Marine	250
Policy Review Screen for Inland Marine	250
Quote Screen for Inland Marine	251
Forms Screen for Inland Marine	252
Payment Screen for Inland Marine	252
Inland Marine Object Model	254
Inland Marine Product Model	256
27 Personal Auto	257
Personal Auto Overview	257
Motor Vehicle Records in Personal Auto	258
Working with Personal Auto	260
Copying Coverages to Other Vehicles	260
Personal Auto Screens	260
Policy Info Screen for Personal Auto	260
Drivers Screen for Personal Auto	262
Vehicles Screen for Personal Auto	263
PA Coverages Screen for Personal Auto	264
Risk Analysis Screen for Personal Auto	265
Policy Review Screen for Personal Auto	265
Quote Screen for Personal Auto	265
Forms Screen for Personal Auto	266
Payment Screen for Personal Auto	266

Personal Auto Object Model.....	269
Personal Auto Object Model Overview	270
PersonalVehicle Entity	270
VehicleDriver Entity.....	271
PolicyDriver Entity	271
Driver Entity	271
Coverages	271
Modifiers.....	272
Motor Vehicle Record Object Model	272
Configuring Personal Auto	275
Configuring Copy Coverages for Personal Auto	275
Configuring Personal Auto Motor Vehicle Records.....	275
28 Workers' Compensation.....	279
Key Features of Workers' Compensation.....	280
Policy Term.....	280
Jurisdictions	280
Workers' Compensation Options	282
Workers' Compensation Screens	283
Qualification Screen for Workers' Compensation	284
Policy Info Screen for Workers' Compensation	284
Locations Screen for Workers' Compensation	286
WC Coverages Screen for Workers' Compensation.....	286
Supplemental Screen for Workers' Compensation.....	289
WC Options Screen for Worker's Compensation.....	289
Risk Analysis Screen for Workers' Compensation.....	291
Policy Review Screen for Workers' Compensation	291
Quote Screen for Workers' Compensation	292
Forms Screen for Workers' Compensation.....	292
Payment Screen for Workers' Compensation	293
Workers' Compensation Object Models.....	295
Workers' Compensation Object Model Overview	296
Policy Contact	297
Coverages	297
Modifiers.....	298
Employees.....	298
Jurisdictions	299
Retrospective Rating Plan.....	299

Part V

Additional Features of PolicyCenter

29 Policy File	303
Policy File Overview	303
Overview of Copying Data Between Policies.....	303
Overview of Split and Spin-off Policies	305
Policy File Screens, Menus, and Actions	306
Info Bar	306
Actions Menu	307
Policy Contract	307
Policy Tools Menu	307
Working with Policies	308
Working with Copy Data	308
Working with Split and Spin-off Policies.....	309

Configuring Policies	309
Core Entities Associated with Policies	309
Policy Object Model Overview	310
Policy Term and Policy Period Entity Relationship Diagram	312
Configuring Copy Data for a Line of Business	313
Configuring Split and Spin-off Policies	314
Split and Spin-off Policies Object Model	314
Split and Spin-off Methods in Gosu Classes	314
30 Archiving in PolicyCenter	315
Archiving Overview	315
Advantages of Archiving	316
Impact of Archiving on Your PolicyCenter Configuration	316
Archiving Policy Terms	316
Policy Terms that PolicyCenter Does Not Archive	317
Entities Retained After Archiving	318
Running the Archive Policy Terms Batch Process	318
Searching for Archived Policy Periods	319
Example of Searching for Archived Policy Periods	319
Desktop and Team Tabs and Archiving	320
Retrieving Archived Policies	320
Example of Retrieving an Archived Policy	320
Configuring Policy Archiving	321
More Information on Archiving	321
31 Account File.....	323
Account Overview	323
Account Security Overview	323
Related Accounts Overview	324
Overview of Moving and Rewriting Policies Between Accounts	324
Overview of Merging Accounts	325
Account Screens	325
Account File Summary Screen	326
Account File Contacts Screen	326
Account File Locations Screen	326
Account File Participants Screen	327
Account File Policy Transactions Screen	327
Submission Manager Screen for Accounts	327
Underwriting Files Screen for Accounts	327
Account File Related Account Screen	327
Account File Documents Screen	327
Account File Notes Screen	327
Account File Claims Screen	327
Billing Screen for Accounts	328
Account File History Screen	329
Account Actions	329

Working with Accounts	329
Searching for an Account	330
Creating an Account	330
Tracking Your Accounts	331
Moving a Policy From One Account to Another	331
Rewriting Policies From One Account to Another	332
Merging Accounts	333
Adding an Account Relationship	334
Modifying an Account Relationship	334
Removing an Account Relationship	334
Searching for Accounts with a Shared Contact	334
Configuring Accounts	335
Account Object Model	335
Account Rule Sets	336
Account Web Service	336
Configuring Moving Policies Between Accounts	337
Moving Policies Between Accounts Data Model	337
Transferring Policies in the Account Plugin	337
Configuring Account Relationships	337
Account Relationship Object Model	337
Account Relationship Typelist	338
Account Relationship Methods in the Account Plugin	339
Account Relationship Rule Sets	339
Account Relationship Methods in Gosu Classes	339
32 Locations	343
Location Overview	343
Types of Location Information	344
Geocoding Locations	344
Synchronizing and Revisioning Location Information	344
Changing Location Information	344
Hiding Location Information on Accounts	345
Location Object Model	346
Account Synchronization Classes for Locations	347
Working with Account Locations	348
Adding a New Location	348
Editing a Location	349
Making a Location Primary	349
Making a Location Active or Inactive	349
Working with Policy Locations	349
Viewing Locations on Bound Policies	350
Editing Location Information on a Policy Transaction	350
Adding or Removing Locations at the Policy Level	350
Location Numbering	350
Configuring Locations	351
Adding a Revised Field to a Location	351
Creating a New Account Location	354
Creating a New Policy Location	354
Configuring Geocoding	354
33 Generic Schedules	357
Generic Schedule Overview	357
Types of Schedules	358

34 Activities	361
Activities Overview	361
Activities Creation and Assignment	362
Activity Ownership	362
Activity Escalation	363
Localizing Activity Patterns	363
Working with Activities	363
Creating an Activity	363
Assigning Activities	363
Completing or Skipping Activities	364
Selecting an Activity from a Queue	364
Activity Patterns	364
Activity Pattern Components	364
Creating and Editing Activity Patterns	366
Using Gosu to Edit Patterns	366
Activity Object Model	367
Activity Batch Process	367
35 PolicyCenter Contacts	369
Contact Overview	369
Centralized View of Contacts on the Contact Tab	370
Sharing Contacts with a Contact Management System	370
Contacts and Roles	370
Types of Contact Information	371
Account Holder Information	371
Revisioning Contact Information in Policies	371
Linking an Address Between Multiple Contacts	374
Working with the Contact Tab	374
Creating a New Contact on the Contact Tab	375
Searching for a Contact on the Contact Tab	375
Selecting Recently Viewed Contacts on the Contact Tab	376
Viewing Contact File Details on the Contact Tab	377
Creating a New Account from a Contact on the Contact Tab	377
Viewing Accounts Associated with a Contact on the Contact Tab	377
Viewing Policies Associated with a Contact on the Contact Tab	377
Viewing Policy Transactions Associated with a Contact on the Contact Tab	378
Viewing Claims Associated with a Contact on the Contact Tab	379
Viewing Billing Information for a Contact on the Contact Tab	379
Viewing the Account Holder Info Screen	380
Contact Tab Behavior	382
Working with Contacts in Policies and Accounts	382
Adding a Contact to an Account	382
Adding a Contact From the Address Book	383
Editing a Contact	384
Removing a Contact from an Account	384
Changing the Active Status on a Contact from an Account	384
Adding or Removing a Contact from a Policy	385
Adding or Removing a Contact Role	385
Working with Linked Addresses	385
Changing Revised Contact Information in a Policy Change	386
Contact Object Model	390
Linked Addresses Object Model	391
Contact Roles for Accounts and Policies	391
Account Synchronization Classes for Contacts	393

Configuring Contacts	394
Data Model Patterns for Contact Roles.....	395
Adding New Contact Roles.....	395
Adding a Revised Field to a Contact.....	399
Configuring Linked Addresses for Contacts.....	405
Gosu Classes for Contacts.....	406
Plugins for Contacts	407
Contact Batch Process.....	407
Configuring the Contact Tab.....	408
36 Underwriting Authority.....	411
Underwriting Authority Overview	411
Specifying Values	412
Checking Sets and Blocking Points	412
Underwriting Referral Reasons on the Policy.....	412
Automatic Approvals	413
Automated Renewals	413
Automated Policy Changes.....	413
Approval Duration	413
History of Underwriting Decisions	413
Working with Underwriting Issues.....	414
UW Issues on the Risk Analysis Screen.....	414
Risk Approval Details Screen	415
Working with Underwriting Referral Reasons	416
37 Quoting and Rating.....	419
Working with Quotes	419
Working with the Quote Screen	419
Entities Associated with Costs and Transactions	420
Cost Delegate	420
Transaction Delegate.....	422
Policy Period Fields for Costs and Transactions	423
Cost and Transaction Model for Business Auto Line	424
Cost and Transaction Model for Businessowners Line	424
Cost and Transaction Model for Commercial Property Line	426
Cost and Transaction Model for General Liability Line	427
Cost and Transaction Model for Inland Marine Line	430
Cost and Transaction Model for Personal Auto Line	431
Cost and Transaction Model for Workers' Compensation Line	433
Calculating Transactions	434
Internal Tools for Rating: Financial Transactions Screen	439
38 Quote Purging	441
Quote Purging Overview	441
Configuration Business Cases	442
What Gets Purged or Pruned?	443
39 Rating Overrides.....	445
Rating Overrides Permissions	445
Underwriting Issues for Rating Overrides	446
Processing Overrides Across Policy Transactions	446
Rating Overrides in the User Interface	447
Rating Overrides on the Quote Screen	447
Rating Overrides Screen	447
Configuring Rating Overrides	448

Adding Rating Overrides to a Line of Business	450
Enabling the Override Rating Button	450
Creating a Panel Set for Rating Overrides	450
Updating the Rating Engine to Handle Overrides.	451
40 Document Management	453
Document Security	454
Hidden Documents	454
Working with Documents	454
Viewing Account and Policy Documents	455
Searching for Documents	455
Adding a New Document	455
Linking to an Existing Document	455
Creating a New Document	456
Editing a Document	456
Removing a Document	456
Using an Activity to Create a Document	456
Creating a Document with a Rule or in a Workflow	457
Relating a Document to a PolicyCenter Entity	457
Creating a Document Template	457
Configuring Document Management	457
Document Management Integration	459
41 Policy Forms	461
Forms Overview	461
42 Policy Data Spreadsheet Import/Export	463
Large Policy Workflow Using Policy Data Spreadsheet Import/Export	464
Policy Data Spreadsheet Import/Export in Commercial Property	464
Using Spreadsheets Generated by Policy Data Spreadsheet Import/Export.	465
Exporting a Spreadsheet from PolicyCenter.	465
Using the Spreadsheets	466
Importing a Spreadsheet	469
43 Product Model Overview	471
Product Model Representation	471
Products Overview	472
PolicyLine Pattern Overview	472
Coverage Pattern Overview	472
Understanding Categories	473
Cov erages, Exclusions, Conditions, and Coverables Overview	473
Existence	474
CoverageTerm Pattern Overview	474
Availability Overview.	475
Determining the Reference Date	475
Making a Pattern Available by Policy Transaction Type	476
Grandfathering Overview	476
Reloading Availability Overview	477
Understanding Offerings.	477
Filtering the Product Model	477
Offering Question Sets	478
Working with Offerings	478

44 Policy Revisioning	479
What Is a Policy Revision?	479
Basic Revisioning Structure of a Policy	481
The Policy Period and Effective Date Fields	485
Structure of Revisioning Across Effective Time	487
Unbound Policy Revisions	489
Slice Mode and Window Mode Overview	489
Slice Mode APIs	490
Window Mode API Overview	491
Safely Accessing Foreign Keys with Slice Mode	493
Version List API Reference	494
Working with Window Mode (Unsliced) Objects	500
Comparing Window Mode Edits to Slice Edits	501
Out-of-sequence Jobs	501
Out-of-sequence Job User Interface	502
Back-dated Versus Out-of-sequence Job	503
Validation Issues and Out-of-sequence Jobs	503
Preempted Jobs	503
Preemption Reduction Configuration	506
Applying Changes to Future Renewals	507
Revisioning Rewrite Jobs	507
Summary of Revisioning Terminology	508
Revisioning Properties Reference	509
Revisioning Properties on a Policy	509
Revisioning Properties on a Policy Period	509
Revisioning Properties on PolicyPeriod Subobjects	511
Details of Merging and Applying Changes	512
Applying Changes Details	513
Merging Changes Details	514
Policy Differences Between Revisions	516
45 Multicurrency Policies	517
Multicurrency Overview	517
Multicurrency Terminology	518
Types of Multicurrency Policies	518
Exchange Rate for Multicurrency Policies	519
Multicurrency and Rating	519
Multicurrency and Reinsurance	521
Multicurrency Object Model	522
Multicurrency Properties	523
Multicurrency in a Policy Line	523
Multicurrency User Interface	524
Multicurrency Fields on Policy Transactions Screens	525
Multicurrency Fields on the Contact Screen	526
Multicurrency Fields on the Account Screen	526
Multicurrency Fields on the Producer Code Screen	526
Multicurrency Field in Reinsurance	527
Multicurrency Fields for Underwriting Authority	527

Part VI Rating Management

46 Rating Management Concepts	531
Rating Management Overview	531
Key Features of Rating Management	532
Rating Management by Line of Business	533
Sample Data for Rating Management	533
Rate Tables	533
Rate Table Overview	533
Rate Table Definition	534
Matching a Factor in the Rate Table	536
Rate Routines	540
Rate Routines That Do Not Calculate Properties on the Cost	541
Rate Routine Versions	541
Rate Routine Variant Identifiers	541
Rate Routine Steps	541
Parameter Sets	542
Rate Books	543
Managing Rate Books and Rate Tables	543
Selecting the Rate Book Edition During Policy Rating	544
Rate Book Matching Process	544
Rate Book Lifecycle and Moving to Production	545
Rating Worksheets	548
Impact Testing	549
47 Rating Management User Interface	551
Preparing to Use Rating Management	551
Working with Rate Books	551
Searching for a Rate Book	552
Adding a New Rate Book	552
Deleting a Rate Book	552
Rate Book Details Screen	553
Rate Book Status and Available Actions	554
Rate Book Actions and Permissions	555
Importing and Exporting Rate Books to XML	556
Working with Rate Table Definitions	557
Searching for a Rate Table Definition	557
Adding a New Rate Table Definition	558
Rate Table Definition Screen	558
Working with the Rate Table Editor	562
Selecting a Rate Table	562
Rate Table Screen	563
Editing Rate Table Content in PolicyCenter	563
Editing Rate Table Content in Excel	564
Rate Table Update Validations	565
Excel Rate Table Import Validations	565

Working with Rate Routines.....	565
Accessing Rate Routines.....	566
Adding a New Rate Routine.....	566
Deleting a Rate Routine	567
Actions on Rate Routines	567
Adding Steps to a Rate Routine	568
Instruction and Operand Types.....	570
Specifying a Function as the Operand in a Rate Routine Step.....	576
Creating a Rate Routine for Another Jurisdiction.....	576
Specifying a Flat-rated Coverage in a Rate Routine.....	576
Viewing Rating Worksheets	577
Working with Parameter Sets	577
Accessing Parameter Sets	578
Adding a Parameter Set.....	578
Adding, Changing, or Removing Parameters from a Parameter Set.....	578
Working with Impact Testing.....	579
Permissions for Impact Testing.....	580
Generating Test Periods with Impact Testing.....	580
Examples of Working with Rating Management	583
Creating and Using a Rate Table with a Multiple Factors	583

Part VII Reinsurance Management

48 Reinsurance Management Concepts	589
Reinsurance Program Basics	590
Using Reinsurance Programs to Serve Business Goals	590
Reinsurance Program Design	591
Reinsurance Program Example.....	592
Reinsurance Agreements	595
Treaties	595
Facultative Agreements.....	596
Proportional Agreements	596
Non-proportional Agreements	599
Summary of Agreement Types.....	603
How PolicyCenter Links Reinsurance to Policies	604
How PolicyCenter Attaches Programs to Policies	604
How PolicyCenter Attaches Agreements to Policies	604
How PolicyCenter Calculates Ceded Premiums.....	607
Calculating Ceded Premiums	608
Ceded Premiums Example	609
Shared Reinsurance Agreements	610
Differential Rates	610
Location Groups	611

49 Reinsurance Management User Interface.....	613
Working with the Reinsurance Tab	613
Searching for Agreements.....	613
Creating a New Treaty	614
Creating a New Program.....	614
Editing a Program	615
Deleting a Program	615
Creating a New Facultative Agreement	615
Validating an Agreement	616
Making an Agreement Active.....	616
Working with Reinsurance Management in Policies	616
Adding Reinsurance to a Policy	616
Creating a Location Group	618
Viewing Ceded Premiums	618
Modifying the Gross Retention.....	619
Adding or Linking to a Facultative Agreement	620
Editing Ceding Parameters	621
Reinsurance Management Screens	621
Treaty or Facultative Agreement Screen.....	621
Reinsurance Program Screen	628
Search Agreements Screen	629
Search Programs Screen	630
Reinsurance Screen in the Policy File.....	630

Part VIII

PolicyCenter Administration

50 Security: Roles, Permissions, and the Community Model	635
Community Model Overview	635
Producers in the Community Model.....	636
Security Overview	637
Role-based Security	637
Data-based Security for Accounts and Policies	638
Data-based Security for the Community Model	642
System and Application Permissions	643
Security Restrictions Using the Status Field.....	643
Producer of Record and Producer of Service	644
Producers of Service Can Edit the Account	644
Changing the Producer	644
Adding a Third Producer.....	644
Managing the PolicyCenter Community	645
Understanding Internal and External Administration.....	645
Security Object Models	646
Object Model for Producer Codes.....	646
Working with Security	647
Viewing Permissions on Selected Roles.....	647
Creating and Removing a Permission.....	647
Adding a Permission to a Role	647
Adding or Removing a Role	648
Turning On Producer Code Security.....	648
Designating a Client Data Integration Handler.....	648

Security and Configuration Scenarios Related to Producer Codes.....	649
Producer Codes Assigned by Level	649
Producer Codes Assigned by Line of Business.....	650
Producer Codes Assigned by Level and Line of Business	650
Producer Codes Roles Customized by User	651
Producer Code Currency.....	651
Security Dictionary.....	651
Access Control for Documents and Notes	652
Working with Access Control for Documents and Notes.....	653
51 Authority Profiles	655
Authority Profile Overview	655
Working with Authority Profiles	657
Viewing or Editing an Authority Profile.....	657
Assigning an Authority Profile to a User	657
52 Team Management	659
Team Management Overview.....	659
Groups and the Team Tab.....	660
Team Tab User Categories	660
Reporting Categories.....	661
Working with the Team Tab.....	662
Assigning Activities	662
Assigning Submissions, Renewals, and Other Policy Transactions.....	662
53 Policy Holds Administration	665
Policy Holds Overview.....	665
Underwriting Holds for Natural Disasters	665
Policy Holds for Regulatory Changes.....	666
Specifying Policy Holds	666
Working with Policy Holds	668
Example: Creating a Simple Policy Hold	668
Working with Policy Hold Actions.....	670
Deleting or Disabling a Policy Hold.....	671
Policy Hold Screen	672
Hold Details Tab.....	672
Hold Regions Tab	673
Copying a Policy Hold	673
Policy Hold Actions in Renewals	673
Policy Hold Object Model	674
Configuring Policy Holds	674
Policy Hold Permissions	675
Policy Hold Authority Grant	675
Policy Hold Underwriting Issue Types	675
Policy Hold Gosu Classes	676
Policy Hold Batch Process	676
Importing and Exporting Policy Holds	677
54 Holidays and Business Weeks	679
Specifying Holiday Dates	679
Holiday Types	680
Working with Holidays, Weekends, and Business Weeks	680
Using Gosu Methods to Work with Holidays	681
Business Weeks and Business Hours	681
Business Hours	682
Holiday Permissions	682

55 Policy Form Pattern Administration	683
About Forms	683
Form Basics	684
Working with Form Patterns	685
Searching for a Form Pattern	685
Adding a Form Pattern	685
Specifying Removal or Replacement Forms for Policy Changes	686
Importing and Exporting Policy Form Patterns	686
Form Pattern or New Policy Form Screen	686
Basics Tab for Form Patterns	687
Products Tab for Form Patterns	687
Transaction Types Tab for Form Patterns	688
Jurisdictions Tab for Form Patterns	688
Policy Change Tab for Form Patterns	690
Inference Tab for Form Patterns	692
Form Configuration	693
Form Pattern Validation	693
Configuring Generic Form Inference	693
Configuring Custom Form Inference	693
Adding a Custom Inference Class	695
56 Administering Policy Data Spreadsheet Import/Export	697
Spreadsheet Export Formats User Interface	697
Defining Export Formats	697

Part IX

Product Integrations

57 Billing System Integration	701
Billing System Integration Overview	702
Organizations and Producer Codes Overview	702
Account Overview	703
Policy Overview	704
Sending Charges and Other Information to Billing	704
Contact Overview	704
Policy Transactions That Create a New Policy Period	705
Policy Transactions That Create Midterm Changes	707
Cancellations in the Billing System Integration	708
Reinstatements in the Billing System Integration	709
Renewals or Rewrites in the Billing System Integration	709
Final Audits in the Billing System Integration	710
Premium Reporting in the Billing System Integration	710
Working with the Billing System Integration	711
Working with the Payments Screen in the Billing System Integration	711
Viewing the Policy Period in BillingCenter	713
Working with Accounts from the Billing System	713
Working with Policies in the Billing System Integration	714
58 Claim System Integration	715
Claim System Integration Overview	715
Accessing Summary Loss Information from the Claim System	716
Viewing Loss Claims for Policies	716
Viewing Loss Claims from an Account	718
Loss Claims Notification at Renewal	718
Approvals	719

Large Loss Notification	719
Permissions for Working with Claims	719
Restricted Fields in Claims	720
Claim Search Plugin	720
59 Contact Management System Integration	721
Contact Management System Integration Overview	722
Creating a New Account	722
Adding Contacts to an Account or Policy.....	722
Updating Contacts.....	722
Searching for Contacts Within a Contact Management System.....	722
New and Updated Contacts.....	723
Pushing New and Updated Contacts.....	723
Adding a Contact from the Contact Management System	724
Creating a New Contact in PolicyCenter	724
Detecting Duplicates in the Contact Management System.....	725
Detecting Duplicates When Integrated with ContactManager.....	725
Duplicate Contacts in PolicyCenter	728
Deleting, Removing, and Inactivating a Contact	729
Customizing the Contact Management System Integration	729
Entry Points into PolicyCenter	730
Integrating with Multiple Contact Management Systems.....	730

About This Document

This application guide describes the functions and processes of Guidewire PolicyCenter. It provides conceptual overviews as well as complete details of how the default application works before configuration.

This topic includes:

- “Intended Audience” on page 25
- “Assumed Knowledge” on page 25
- “Related Documents” on page 25
- “Conventions in This Document” on page 27
- “Support” on page 27

Intended Audience

This guide is meant as an introduction to PolicyCenter. It is a good starting point for

- Implementation team members or IT staff who seek a better understanding of how PolicyCenter works and what is contained in the default application
- Business analysts who define the business logic
- Anyone who has a vested interest in understanding PolicyCenter

Assumed Knowledge

This guide is your starting point in the PolicyCenter documentation and introduces you to concepts that are the basis of the application. Prior knowledge of PolicyCenter is not required. See the following topic to see which documents to use for implementing and configuring PolicyCenter.

Related Documents

For details on the various aspects of PolicyCenter, see the following:

PolicyCenter New and Changed Guide – Describes new features and changes to existing features in this version of PolicyCenter.

PolicyCenter Upgrade Guide – Provides instructions to upgrade PolicyCenter.

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

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

PolicyCenter Configuration Guide – Describes how to configure PolicyCenter and includes basic steps and examples for implementing such configurations. This guide is intended for IT staff and system integrators who configure PolicyCenter for an initial implementation or create custom enhancements. Use this guide as a general reference.

PolicyCenter Rules Guide – Describes the business rule methodology, rule categories for PolicyCenter, and rule syntax for Guidewire Studio. This book is intended for programmers who write Gosu business rules and analysts who define the business rule logic.

Guidewire Contact Management Guide – Describes how to install and configure ContactManager and how to integrate it with *PolicyCenter*. Includes basic steps and examples. This guide is intended for IT staff and system integrators who configure ContactManager for an initial integration or who create custom enhancements.

Gosu Reference Guide – Describes the syntax of Gosu expressions and statements within PolicyCenter. This document also provides examples of how the syntax is used when creating rules, specifying sources of information in page configurations, and writing query builder expressions.

PolicyCenter Gosu Generated Documentation – Documents all types visible from the Gosu type system. This includes Guidewire entities, Gosu classes, utility classes, Gosu plugin definitions, and Java types that are available from Gosu. With a local copy of the product, you can regenerate this documentation. Go to the `PolicyCenter/bin` directory and run the `gwpc regen-gosudoc` command.

PolicyCenter Integration Guide – Provides an architectural overview and examples of how to integrate PolicyCenter with external systems and custom code. This document is a learning tool for explanations and examples, which supplements the *Java API Reference Javadoc* and *SOAP API Javadoc*. This document is written for integration developers and consultants.

PolicyCenter Glossary – Defines terms used in the documentation.

PolicyCenter Data Dictionary – Describes the PolicyCenter data model, including your data model extensions. To generate the dictionary, go to the `PolicyCenter/bin` directory and run the `gwpc regen-dictionary` command. To view the dictionary, open the `PolicyCenter/build/dictionary/data/index.html` file. For more information about generating and using the *Data Dictionary*, see “Working with the Data Dictionary” on page 143 in the *Configuration Guide*.

PolicyCenter Security Dictionary – Documents security permissions, roles, and the relationships between them. Generate the dictionary by going to the `PolicyCenter/bin` directory and running the `gwpc regen-dictionary` command. To view the dictionary, open the `PolicyCenter/build/dictionary/security/index.html` file. See “Regenerating the Data Dictionary and Security Dictionary” on page 30 in the *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 .
<code>monospaced</code>	Literal text that you can type into code, computer output, class names, URLs, code examples, parameter names, string literals, and other objects that might appear in programming code. In code blocks, bold formatting highlights relevant sections to notice or to configure.	Get the field from the <code>Address</code> object.
<code>monospaced italic</code>	Parameter names or other variable placeholder text within URLs or other code snippets.	Use <code>getName(first, last)</code> . <code>http://SERVERNAME/a.html</code> .

Support

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

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

part I

Introduction

Introduction to PolicyCenter

PolicyCenter is a web-based underwriting and policy administration system designed for personal and commercial line carriers in the property and casualty insurance (P&C) industry. In PolicyCenter, producers and underwriters can submit applications, renew policies, and manage policy changes. Auditing is available for certain types of policies. PolicyCenter provides access to agents, and supports producer relationships and underwriting risk assessment. Typical users, such as underwriters and producers, can create and manage policies, service accounts, evaluate risks, view policies, create activities, and handle inquiries. PolicyCenter also provides access management tools for viewing groups and repurposing workloads.

PolicyCenter stores information about a policy and manages a set of processes that, if completed successfully, result in changes to the policy. Examples of policy changes are: creation of a new policy, renewal of a policy for a new term, or cancellation of a policy. As a result of each policy transaction (such as adding an additional driver to an auto policy), the system determines the price of the transaction. If successfully completed, PolicyCenter forwards this pricing information to a billing system. The pricing information is also important for reporting to regulators.

In PolicyCenter, you can tailor your products. Before you can use PolicyCenter to manage policies, you must first define your product line. In other words, what products are you going to offer? Products are the first level of the product model hierarchy. Carriers or agents sell these products (such as personal auto or workers' compensation policies) to customers. Each individual policy is an instance of a product. Therefore, personal auto, business-owners, and workers' compensation are all products.

While PolicyCenter comes with certain lines of business, its real power is in its flexibility. You can customize the default lines of business to meet your business needs. You can also create your own lines of business in Guidewire Studio. (PolicyCenter includes Studio.)

How do you configure PolicyCenter? Use Guidewire Studio as the integrated development environment (IDE) to configure PolicyCenter to meet your business needs. In Studio, you can control workflow, policy transactions, PCF screens, typelists, rules, and Gosu.

How are lines different from products? For example, general liability and commercial property are both lines. If you represent a business, you can buy a general liability product, which just includes the general liability line. Then you can buy a commercial property product, which includes just the commercial property line. However, you can also buy a commercial package product, which may include both the general liability and commercial property lines. Commercial package is a multi-line product, as opposed to a monoline product. Conversely, an

insurer can sell multiple products, each of which includes the same line or lines. A product can be targeted to a particular group. For example, an insurer can offer a commercial package for shopping centers, one for hotels, and one for universities.

How do you manage policies? You create and manage policies through a web interface. On this virtual **Desktop**, you create policy transactions (or jobs) that process policies in various ways. Through policy transactions, you can submit, issue, change, renew, cancel, reinstate, rewrite, and audit policies.

Types of policy transactions include: submission, issuance, policy change, renewal, cancellation, reinstate, rewrite, and audit.

This topic includes:

- “Product Model” on page 32
- “The Policy Lifecycle” on page 33
- “PolicyCenter Integration Points” on page 34
- “Integration with Other Guidewire Applications” on page 35
- “PolicyCenter Users” on page 36

Product Model

The PolicyCenter product model is at the core of its line of business configuration. It defines the products that carriers offer through PolicyCenter. PolicyCenter stores these product definitions as patterns. PolicyCenter uses these patterns during the submission process to generate instances of policies or the subcomponents of policies. Use Guidewire Studio to create and manage your product model.

Envision the product model as a product configurator. You can equate it to a computer manufacturer’s web site that offers several models of laptop computers, each with different features to select from. Manufacturers typically design their product for a specific market. The manufacturer provides one laptop for consumers who travel and need a light weight laptop. They provide another for consumers who need a very fast computer with large amounts of RAM and the latest graphics card. The manufacturer figures out what consumers want and creates products that fill that need.

You can configure PolicyCenter to meet your business needs. You can define a product, select a policy line, and offer different coverages and coverage terms.

PolicyCenter is dynamic – you can define, create and implement products within weeks. For each product, the product model defines what each product can cover. For example, an auto policy includes information about collision coverage and uninsured motorist property damage coverage. You configure the business logic in Studio.

Example

Janet Jones, an Acme Insurance producer, has noticed many people driving the new hybrid auto (gas and electric). She has an idea that she thinks her manager will like. So she begins to research the idea of offering a new coverage that can bring more revenue to the company. Her extensive research indicates that more than 20% of new car drivers in three surrounding cities drive the new vehicle. Further research also indicates that the life span of the battery is about 80,000 miles. She proposes to her manager that Acme can be the first insurance company in their area to offer special coverage on the hybrid that would cover batteries. Her manager likes the proposal. Offering a new hybrid coverage can be implemented in PolicyCenter in weeks, instead of the years that it would take an IT department to make changes on a legacy system. This speed gives insurance companies the competitive edge to get to market quickly.

Lines of Business in the Base Application

The PolicyCenter base application contains several lines of business. Each line of business contains a reference implementation that you can use to accelerate your implementation. Each line of business includes reference

implementations for policy transactions, policy file screens, and sample rating rules. A line of business may contain forms and forms logic. A line of business may also contain rules to determine eligibility. Each reference implementation also provides sample content, though the extent of this content varies by line of business.

PolicyCenter includes the following lines of business:

- Businessowners
- Commercial Auto
- Commercial Property
- General Liability
- Inland Marine
- Personal Auto
- Workers' Compensation

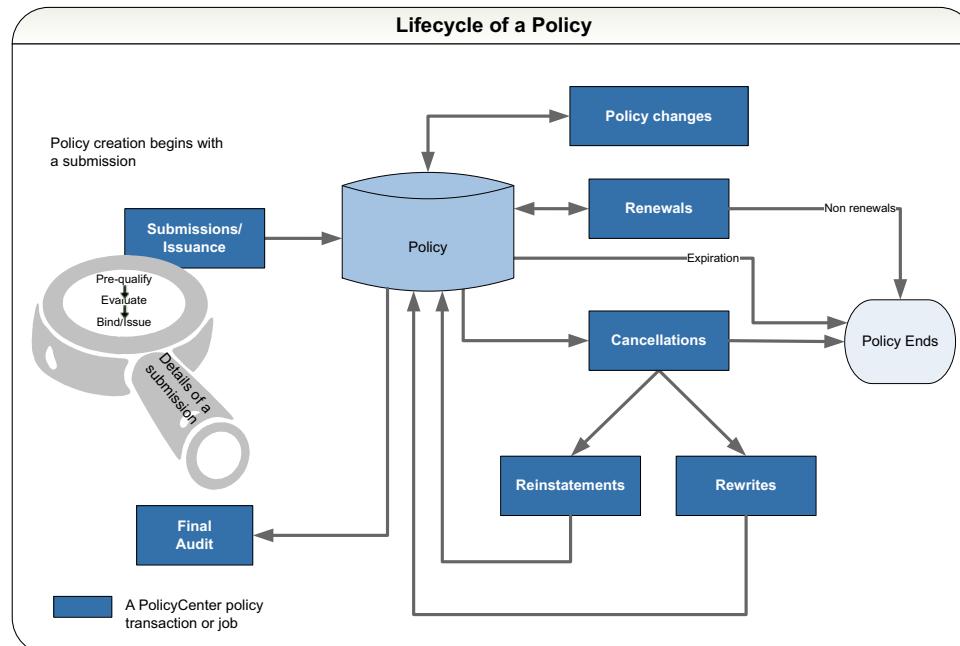
Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can maintain your lines of business within your jurisdictions, as necessary. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. Lines of business can accommodate many coverages and exclusions per jurisdiction over time. The default application contains a sample set of these classifications, coverages, and exclusions.

There is a corresponding product for each line of business. In addition, Commercial Package is a multi-line product that includes Commercial Property, Inland Marine, and General Liability lines of business.

The Policy Lifecycle

The core of PolicyCenter revolves around the policy. So it is helpful to understand the lifecycle of a policy, which includes policy transactions, within PolicyCenter.

Note: This diagram does not attempt to display all the details in each policy transaction, but rather provides a high level view. You can find detailed descriptions for each policy transaction in subsequent topics.



Submissions

The goal of the submission process is to create a policy and have the policyholder accept it. After entering the policyholder's information, the producer gives a quote. If the policyholder agrees and accepts it, then the producer binds the policy and sends it out with the accompanying documentation. The producer also forwards the billing information to an external billing system (not shown in the diagram).

Policy Changes

Any changes to a policy can require additional evaluation on the part of an underwriter and result in a change to the premium. A typical change might include additions to the policy (such as adding drivers or cars) or changes to coverage limits and deductible amounts.

Renewals

The normal progression just before a policy expires is to renew it for another period of time – six to 12 months is typical. After PolicyCenter renews a policy, it returns the policy to maintenance mode until the policy changes, expires, cancels, or renews again.

Cancellations and Reinstatements

You can also cancel policies. Before the cancellation processes completes, a cancellation can be rescinded. An example is a producer mailing a cancellation notice for non-payment to a policyholder. If the policyholder corrects this by submitting payment before the cancellation date then the cancellation can be rescinded with no break in coverage.

Reinstatements

Reinstatements go hand in hand with cancellations and are a type of policy change that returns a canceled policy to in-force status. The policy is in-force as of the reinstatement date. The reinstatement removes the cancellation from the policy period since the period is no longer canceled. The expiration date remains the same.

Rewrites

When there are many errors are on a policy, it becomes necessary to rewrite it. Policies must first be canceled before being rewritten.

Audits

The audit policy transaction lets the carrier verify information about the policyholder so that they can determine the accuracy of premiums paid. The audit policy transaction provides final audit and premium reports.

PolicyCenter supports only final audit for the workers' compensation line of business. You set up the method of final audit (physical, voluntary, or by phone) when you create the workers' compensation policy.

With premium reports, the policyholder is billed for premium based on periodic requests for actual basis amounts, such as payroll. A deposit, usually a percentage of the estimated annual premium, is billed at the beginning of the policy. As each reporting period ends, the policyholder is billed based on the actual basis reported by them.

PolicyCenter Integration Points

By design, PolicyCenter is flexible and can integrate with many applications and services. These integration points need to be considered as you configure the application. Some are mandatory while others are optional, depending on your business needs. The types of systems that you may integrate PolicyCenter with include:

- **Legacy Policy Administration systems** – As users renew or change policies, you can import the policy data into PolicyCenter.

- **Billing system** – When a user creates a policy, PolicyCenter exports billing information. PolicyCenter also sends and receives information as the policy changes.
- **Claims system** – Information is sent to and from claim systems. The claims system can send information about the number and type of claims against a policy. PolicyCenter sends policy data to the claims system, such as policy effective dates which answer the question, “Was the policy in-force when this accident occurred?”
- **Print Issuance system** – This system produces policy forms and letters that need to be printed (issued).
- **Document Storage system** – This system stores documents that need to be tracked in a central repository.
- **Database Warehouse/Reporting system** – Use this system for reporting purposes.
- **Authentication system** – Users may need to be authenticated from other systems.
- **Contact Management or Address Book application** – It is often necessary to store and maintain contact information separately from PolicyCenter because users outside of PolicyCenter may need access to it. For information on how PolicyCenter integrates with a contact management system, see “Contact Management System Integration” on page 721.
- **Rating Engine** – The system that rates a policy and sends the quote information back to PolicyCenter. It also calculates the estimated annual premium in audits.
- **Department of Motor Vehicles** – When you integrate with this system, agents can request and send driver information to the regulatory body.
- **VIN (Vehicle Identification) Service** – Use this system to look up VIN information. The default configuration contains a demo plugin.
- **Producer Management System** – This central repository stores producer related information, such as producer codes used in territories, and information about producers, such as their licensing.
- **Vendor Management system** – Carriers use a vendor management system to track vendor related information that a carrier uses. An example of a vendor might be an outside audit company.
- **Sales Portal or application** – Some carriers may choose to separate the process of collecting account information and submission proposals from the actual issued policy.
- **Actuarial/Statistical system** – PolicyCenter can send data to a system for actuarial analysis.
- **State Insurance Bureaus/Department of Insurance** – This legal entity performs a number of duties including tracking follow-up information for each policy. This entity also sends back suggested rates and experience modification information. PAS systems may also send their proof of insurance.
- **Address Normalization and Validation services** – Provides normalization and validation against information provided by the United States Postal Service (USPS). The USPS provides standardized abbreviations such as *ste* for suite and *ln* for lane. It also lists the complete range of numeric addresses per street, street addresses per ZIP code, ZIP codes per state.
- **Credit Rating system** – A policy administration system may periodically request information about an insured. An example of a credit rating system is Dun and Bradstreet.

For more information on how to integrate with other systems, see the “Integration Overview” on page 25 in the *Integration Guide*.

Integration with Other Guidewire Applications

The default installation of PolicyCenter provides integrations with:

- Guidewire BillingCenter
- Guidewire ClaimCenter

These integrations are easily enabled. You can customize these integrations for your business needs.

See also

- “Billing System Integration” on page 701
- “Claim System Integration” on page 715

PolicyCenter Users

There are several types of users in PolicyCenter. Typically, users spend much time working on policy transactions or looking up a policy’s status to answer questions. Looking up information is relatively simple: users search for an account or a policy and view available data through the user interface. Managing policy transactions is more complex. Users initiate some transactions (for example, an agent fills out a submission to get a quote). Other transactions are a mix of automated and manual handling. For example, renewals are usually automated, but are sometimes referred to an underwriter. PolicyCenter also supports activities, notes, attached documents, history, team views, and more to help users keep track of their work, collaborate with others, and keep these processes moving.

The following table lists typical PolicyCenter users and their roles:

Users	Typical Activities
Agents (independent, captive, or direct)	<ul style="list-style-type: none">• Answer queries regarding policies• Submit a quote• Change or cancel a policy
Policy Service Reps Policy Processors	<ul style="list-style-type: none">• Data entry and tracking policies• Answer simple queries
Underwriters	<ul style="list-style-type: none">• Review accounts and policies• Review policy changes• Rewrite policies• Evaluate (do risk analysis on) policies and policyholders
Underwriting Management	Use team screens to: <ul style="list-style-type: none">• Monitor work loads of subordinates• View subordinate activities• View policies assigned to subordinates
Claims staff	<ul style="list-style-type: none">• View policies• Leave notes or attach documents to accounts and policies• Send messages (such as a risk alert) indicating that a policy has a large loss
Accounting staff	<ul style="list-style-type: none">• View policies• Leave notes or attach documents to accounts and policies• Send messages (such as a non-renewal alert) recommending to not renew a policy due to non-payment
Auditors (internal and external)	<ul style="list-style-type: none">• Audit policies• Provide input to underwriters

part II

PolicyCenter User Interface

Navigating PolicyCenter

This topic describes how to access PolicyCenter and provides instructions on how to navigate the user interface.

This topic includes:

- “Logging into PolicyCenter” on page 39
- “Setting Preferences” on page 40
- “Selecting International Settings in PolicyCenter” on page 40
- “Common Areas in the PolicyCenter User Interface” on page 42
- “PolicyCenter Tabs” on page 43

Logging into PolicyCenter

To log into PolicyCenter, you need:

- **A web browser** – For information about supported browsers, see “Client Information” on page 39 in the *Installation Guide*.
- **The URL (web address) for connecting to PolicyCenter** – Contact your system administrator for details on installation and the web address to use. You can set up a **Favorite** link to the URL or a create a shortcut on your computer desktop that starts a web browser with that URL.
- **A user name and password**

1. Launch PolicyCenter by opening up an instance of a web browser, entering the appropriate web address, such as:

`http://localhost:8180/pc/PolicyCenter.do`

2. Enter your **User Name** and **Password** on the login screen.

If you first click **Keep me logged in** and then log in, you are logged in automatically for the next seven days whenever you navigate to the login page. To support this feature, the application must be hosted by the same application server each time you return to the page. If you log out of PolicyCenter, you will need to log in the next time you navigate to the login page.

When you click **Keep me logged in**, PolicyCenter writes a cookie to your machine. As is the case with all Internet Explorer cookies, writing this cookie can expose a security risk if other people get access to the cookie. For example, someone could copy the cookie to another machine and then be able to log in without entering a user name or password. If this security issue is a concern, you can remove this field from the login page in your PolicyCenter implementation.

When you log in, PolicyCenter displays your startup view (or landing page).

Since PolicyCenter generates screens dynamically:

- You cannot create **Favorites** to screens other than the login screen.
- The **Back** button of the browser is not supported.

In the default configuration, PolicyCenter opens to the **My Activities** screen on the **Desktop** tab. This screen lists all open activities that have been assigned to you.

Setting Preferences

You can set user preferences by clicking the **Preferences** link in the upper right corner of the **Tab Bar**. Your changes take effect the next time you log in.

In the **Preferences** worksheet you can:

- Request email notification when an activity is assigned to you. In the default configuration, selecting this item does not enable email notification. This feature must be configured.
- Change the page that PolicyCenter displays when you log in by selecting a **Startup Page**.
- Change your password.
- Determine how many recent accounts, policies and policy transactions, or contacts display at the end of the **Account**, **Policy**, and **Contact** tab menus. For each user, the recently viewed list is initially empty. Accounts, policies and policy transactions, and contacts are added as the user views these items over multiple sessions. More recently viewed items appear higher on the list. Once the maximum number of recent items has been reached, older items are removed and replaced by newer ones.

The value of this field must be between 1 and 10, inclusive.

If the field has no value, PolicyCenter uses a value in **Other Resources → config.xml** in Studio. In the default configuration, the parameter value is 5. The value can be between 1 and 10, inclusive. Other values generate an error when PolicyCenter starts. The parameters in config.xml are:

Preferences label	config.xml parameter
Maximum Recent Accounts	MaxRecentAccounts
Maximum Recent Policies And Policy Transactions	MaxRecentPoliciesAndJobs
Maximum Recent Contacts	MaxRecentContacts

Selecting International Settings in PolicyCenter

Within the Guidewire PolicyCenter application, each user can set the following:

- The language in which PolicyCenter displays labels and drop-down menu choices
- The regional formats that PolicyCenter uses to enter and display dates, times, numbers, monetary amounts, and names.

You set your personal preferences for display language and for regional formats by selecting **International** from the Options  menu at the top, right-hand side of the PolicyCenter screen. Then, you select options for language and

regional formats independently. From the Options  menu, select **International**, and then select one of the following:

- **Language**
- **Regional Formats**

To take advantage of international settings in the application, you must configure PolicyCenter with more than one locale.

- PolicyCenter hides the **Language** submenu if only one language is installed.
- PolicyCenter hides the **Regional Formats** submenu if only one locale is configured.
- PolicyCenter hides the **International** menu option entirely if a single language is installed and PolicyCenter is configured for a single locale.

PolicyCenter indicates the current selections for **Language** and **Regional Formats** by placing a check mark to the left of the selected options and displaying them in gray.

Options for Language

In the base configuration, Guidewire has a single display language, English. To view another language in PolicyCenter, you must install a language module and configure PolicyCenter for that language. If your installation has more than one language, then you can select among them from the **Language** submenu. The **LanguageType** typelist defines the set of language choices that the menu displays.

If you do not select a display language from the **Language** submenu, then PolicyCenter uses the default language. The configuration parameter **DefaultApplicationLanguage** specifies the default language. In the base configuration, the default language is English.

Options for Regional Formats

If your installation contains more than one configured locale, then you can select a regional format for that locale from the **Regional Formats** submenu. At the time you configure a locale, you define regional formats for it.

Regional formats specify the visual layout of the following kinds of data:

- Date
- Time
- Number
- Monetary amounts
- Names of people and companies

The **LocaleType** typelist defines the names of regional formats that users can select from the **Regional Formats** submenu. The base configuration defines the following locale types:

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

Before you select a regional format from the **Regional Formats** submenu, PolicyCenter uses the regional formats of the default locale. The configuration parameter **DefaultApplicationLocale** specifies the default locale. In the base configuration, the default locale is **United States (English)**. After you select your preference, you cannot revoke your choice and use the default locale again. Instead, you must select the default locale as your personal preference.

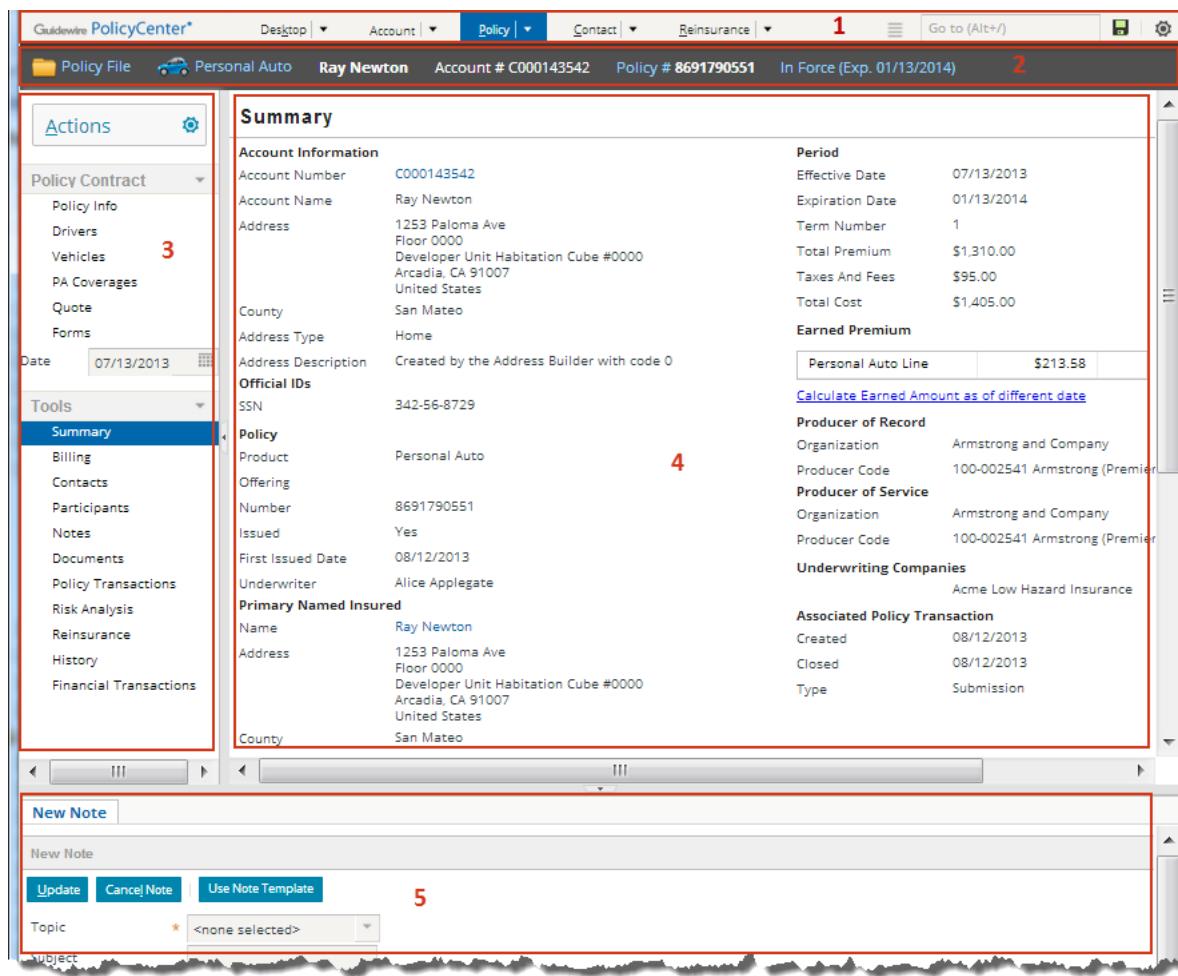
See also

- “Understanding Globalization” on page 9 in the *Globalization Guide*

- “Language Display in Guidewire PolicyCenter” on page 23 in the *Globalization Guide*
- “Working with Localization” on page 17 in the *Globalization Guide*

Common Areas in the PolicyCenter User Interface

This topic provides descriptions of common areas of the PolicyCenter user interface.



The PolicyCenter main user interface contains the following areas:

Area	Description
1	<p>The Tab bar contains:</p> <ul style="list-style-type: none"> • Tabs • QuickJump box is the text box that displays Go to (Alt+J). For more information, see “QuickJump” on page 51. • Unsaved Work menu. For more information, see “Saving Your Work” on page 67. • Options menu contains various links including International, Help, About, Preferences, and Log Out.
2	The Info bar displays information pertaining to your immediate task as seen in the main screen area (#4). The Info bar is not always visible. In the base configuration, the Info bar is visible only on the Account and Policy tabs. The Info bar may have links that allow you to navigate up a level, such as from a policy to an account.

Area	Description
3	The Sidebar contains menu links and the Actions menu. Use the Sidebar menu links to navigate to different pages. The items in the Sidebar are contextual and change depending on the policy object. The Date field displays the <i>as of</i> date of the policy term. PolicyCenter displays the policy data effective as of this date.
4	The Screen Area displays most of the business information.
5	The Workspace can display information separate from the Screen Area, such as modifying your Preferences or viewing or adding a note.

PolicyCenter Tabs

In PolicyCenter, tabs group together logical functions. Tabs can also contain menus with shortcuts to screens on that tab. To see these menus, click the down arrow next to the tab name and select the link from the drop-down menu.

PolicyCenter has these main tabs:

- “Desktop Tab” on page 43
- “Account Tab” on page 46
- “Policy Tab” on page 47
- “Contact Tab” on page 47
- “Reinsurance Tab” on page 48
- “Search Tab” on page 48
- “Team Tab” on page 49
- “Administration Tab” on page 49

Desktop Tab

The **Desktop** tab is the electronic desktop that organizes the user’s activities, accounts, and other items. In the left sidebar and from the **Desktop** drop-down menu, the **Desktop** tab has links to the following screens:

- **My Activities**
- **My Accounts**
- **My Submissions**
- **My Renewals**
- **My Other Policy Transactions**
- **My Queues**

These screens have search drop-down lists that filter activities, accounts, and other items related to the current user.

My Activities

The **My Activities** screen displays activities that have been assigned to you. With activities, you can track tasks associated with an account, policy, or policy transaction.

You can reassign an activity to another user, skip an activity, or mark an activity as complete. Skipping an activity indicates that you no longer wish to do the activity. Completing an activity marks it as finished.

On the **My Activities** screen, users see the following items in the search drop-down list:

- **All open** – Display all activities assigned to the current user.
- **My activities today** – Display all activities for the current user that are due today.

- **Due within 7 days** – Display all activities for the current user that are due within seven days.
- **Overdue only** – Display all activities for the current user that are overdue.
- **Closed in last 30 days** – Display all activities for the current user that were closed in the last 30 days.

See also

- “Activities” on page 361

My Accounts

The **My Accounts** screen displays accounts that you recently created or are working on. Click an account number link to go directly to the **Account File Summary** screen in the **Account** tab for that account.

To view this screen, the user must have the **View my accounts** permission. The code for this permission is `viewmyaccounts`. In the default configuration, the **Producer**, **Producer Code - Basics**, and **Superuser** roles have this permission.

Users see the following items in the search drop-down list:

- **All pending** – Display all pending accounts on which the current user has a role.
- **Created in past 7 days** – Display all account created in the past seven days on which the current user has a role.

The **UserRoleAssignment** object, accessed through `Account.RoleAssignments` array, contains the users with roles on the account. An account is pending if `Account.AccountStatus` is `Pending`. The `Account.CreateTime` property is used to determine whether the account was created in the past seven days.

See also

- “Account File” on page 323

My Submissions

In a submission, you can gather information for binding and issuing a policy. For certain user types, the **My Submissions** screen displays submission or issuance policy transactions that the user created or is working on. For other user types, the screen displays submissions that the user is associated with through an activity. This is similar to how the **Team** screen displays policy transactions depending upon whether the user is a *by-role* or *by-activity* user. For more information, see “Team Tab User Categories” on page 660.

Use the search drop-down list to filter your search. Filters expected to return the largest number of results are only available with certain permissions. Permissions limit the search results for users involved in many different submissions.

To view this screen, you must have the **View my submissions** permission. The code for this permission is `viewmysubmissions`. Users with the **View producer style desktop details** permission see additional items in the search drop-down list. In the default configuration, the **Producer**, **Producer Code - Basics**, and **Superuser** roles have the **View producer style desktop details** permission. The code for this permission is `viewproducerstyledesktopdetails`.

By-role users have the **View my submissions** and **View producer style desktop details** permissions. These users see the following items in the search drop-down list:

- **Open with activity for me** – Display open submissions and issuance policy transactions for which the current user is assigned to an open activity.
- **Open with activity for me due within 7 days** – Display open submissions and issuance policy transactions for which the current user is assigned to an open activity that is due within 7 days.
- **Open bound** – Display the current user’s open issuance policy transactions.
- **All open** – Display all open submissions for the current user.
- **Created in past 7 days** – Display all submissions that the current user created in the past 7 days.
- **Completed in last 30 days** – Display all submissions that the current user completed in the last 30 days.

By-activity users have the **View my submissions** permission but not the **View producer style desktop details** permission. These users see the following items in the search drop-down list:

- Open with activity for me
- Open with activity for me due within 7 days
- Open bound

The **View producer style desktop details** permission affects which columns the user sees. A user who has the permission, sees columns relevant to the policy's producer. For example, this user sees the **Underwriter** column which displays the submission's underwriter. This information is relevant to a producer.

A user who does not have this permission, sees columns relevant to a user who is not the policy's producer, such as the underwriter. For example, this user sees the **Producer** column which displays the submission's producer. This information is relevant to an underwriter.

A user is related to a submission if one of the following is true:

- If the user has a **UserRoleAssignment** for the policy transaction.
- If an activity on the policy transaction is assigned to the current user, and the activity has been modified within **SearchActivityThresholdDays** before the current date. The **Activity.UpdateTime** field contains a timestamp of when the activity was last modified.

See also

- “Submission Policy Transaction” on page 79 and “Issuance Policy Transaction” on page 89
- “**SearchActivityThresholdDays**” on page 46 in the *Configuration Guide*

My Renewals

The **My Renewals** screen displays renewals you recently created or are working on.

To view this screen, the user must have the **View my renewals** permission. The code for this permission is **viewmyrenewals**. Users with the **View producer style desktop details** permission see additional items in the drop-down list.

The items in the search drop-down list that the user sees are similar to the ones on the **My Submissions** screen but apply to renewal policy transactions. This screen does not have the **Open bound** item. For descriptions of these items, see “My Submissions” on page 44.

See also

- “Renewal Policy Transaction” on page 93

My Other Policy Transactions

The **My Other Policy Transactions** screen displays other policy transactions you created or are working on.

To view this screen, the user must have the **View my policy changes** permission. The code for this permission is **viewmypolicychanges**. Users with the **View producer style desktop details** permission see additional items in the drop-down list.

The items in the search drop-down list that the user sees are similar to the ones on the **My Submissions** screen. The filter applies to policy change, cancellation, reinstatement, renewal, rewrite, rewrite new account, and audit policy transactions. This screen does not have the **Open bound** item. For descriptions of these items, see “My Submissions” on page 44.

See also

- “Policy Change Transaction” on page 115
- “Cancellation Policy Transaction” on page 107

- “Reinstatement Policy Transactions” on page 125
- “Renewal Policy Transaction” on page 93
- “Premium Audit Policy Transaction” on page 143

My Queues

The **My Queues** screen displays activities that have been assigned to groups you belong to, but have not been assigned to a specific individual.

To view this screen, you must have the **View my queues** permission. The code for this permission is `viewmyqueues`.

Click **Assign Next to Me** to assign the activity to yourself and remove it from the queue. A queue is a repository which contains activities assigned to a group but not to a particular user in that group. For more information, see “Selecting an Activity from a Queue” on page 364.

Configuring Search Drop-down Lists

You can configure the search drop-down lists on the **Desktop** tab screens. The changes you can make to the functionality include:

- Adding or removing list items
- Changing the functionality that determines what the list item displays
- Adding or changing user permissions that control list item visibility

The **My Activities** screen provides a simple example.

To view or modify the search filters for the My Activities screen

1. In Studio, open the `DesktopActivitiesLV` PCF file.
2. Select the `activitiesFilter ToolbarFilter` PCF element.
3. At the bottom of the screen, click the **Filter Options** tab.
4. Select `web.desktop.DesktopActivityFilters.filters()`.

The filter is defined by `web.desktop.DesktopActivityFilters.filters()`.

5. Open `DesktopActivityFiltersEnhancement.gsx` in the `gw.api.filters` package.

You can view or modify the code that filters the drop-down list items on the **My Activities** screen.

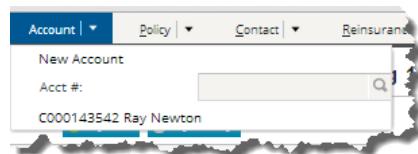
See also

- “Filtering Results with Standard Query Filters” on page 160 in the *Gosu Reference Guide*

Account Tab

From the **Account** tab, you can either create a new account or find an established one. If you select the **Account** tab directly, PolicyCenter displays accounts that you recently worked on at the bottom of the drop-down menu.

Select an account to display that account information in the Account File. The Account File includes information about the account itself, its contacts and locations, the policies held by the account, and policy transactions (such as submissions and renewals) for the account.

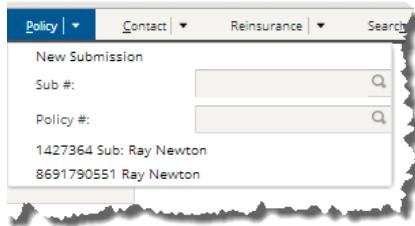


You can edit account information, or change the account holder to another person or company. To learn about managing account information, see “Account File” on page 323.

For information about setting the number of recent accounts that PolicyCenter displays on the **Account** tab, see “Setting Preferences” on page 40.

Policy Tab

Like the **Account** tab, the **Policy** tab remembers the last few policies you worked on. Clicking on **Policy** takes you directly to the policy file for the last policy you worked on. The policy file includes both the policy contract information and the policy tools information. The policy contract describes what the policy covers. The policy tools provide supporting information about the work done on the policy, such as notes, documents, workplan, and risk analysis.



You can also do the following:

- Create a submission.
- Find a submission or policy.

To learn about the policy file, see “Policy File” on page 303.

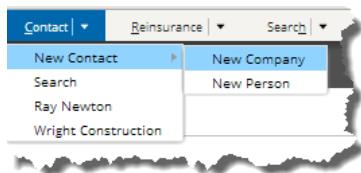
For information about setting the number of recent policies and policy transactions that PolicyCenter displays on the **Policy** tab, see “Setting Preferences” on page 40.

Contact Tab

The **Contact** tab provides a central place to view information associated with a contact such as:

- Details including name, phone, date of birth, addresses, and other information
- Accounts
- Policies
- Work orders
- Claims if PolicyCenter is integrated with claims system
- Billing if PolicyCenter is integrated with a billing system

Using the **Contact** tab, you can create new contacts, search for existing contacts, or select a recently viewed contact. You can also create an account for the contact.



See also

- “Working with the Contact Tab” on page 374
- “Configuring the Contact Tab” on page 408

- “Setting Preferences” on page 40 for information about setting the number of recent contacts that PolicyCenter displays on the Contact tab.

Reinsurance Tab

The Reinsurance tab is accessible if you have Guidewire Reinsurance Management enabled. Reinsurance Management is available within PolicyCenter. However, Reinsurance Management is licensed separately from PolicyCenter.

Use the Reinsurance tab to view and define reinsurance agreements and programs.

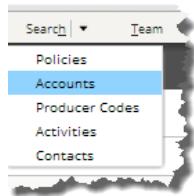
See also

- “Reinsurance Management Concepts” on page 589

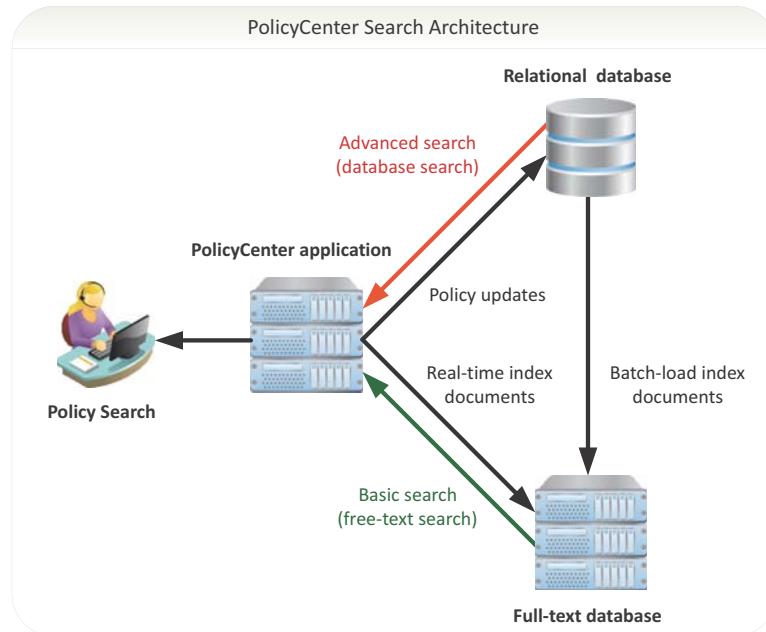
Search Tab

Use the Search tab to find:

- Policies
- Accounts
- Producer Codes
- Activities
- Contacts



PolicyCenter includes two types of searches: basic and advanced.



Basic search is a free-text search for quick access against very large databases. Free-text search also provides exact and inexact matching. Inexact matching returns results that partially match, are synonyms, and sound-like the search criteria. In PolicyCenter, free-text search uses an integration with the full-text search engine Solr. PolicyCenter includes basic search for policies.

Advanced search uses database search, which directly searches the PolicyCenter database. PolicyCenter includes advanced search for policies, accounts, producer codes, activities, and contacts. For large data sets, advanced search can take longer than basic search.

See also

- “Basic Search” on page 55
- “Advanced Search” on page 63

Team Tab

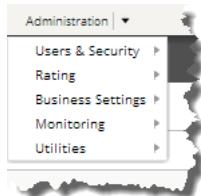
In the PolicyCenter Team tab, supervisors and managers can manage their teams, obtain instant status information, monitor case loads, identify backlogs, and reassign activities. In some respects, this tab serves as a reporting tool. For example, a supervisor can see real time summaries of activities based on groups, then navigate to view and manage a subordinate’s workload.

The Team tab has no drop-down menu choices.

To learn about team management, see “Team Management” on page 659.

Administration Tab

Certain users with assigned roles, such as producers, can use the Administration tab. This tab contains menu items to search for users, organizations, or producer codes. These users can search for information about the carrier and see the carrier’s organization. The permissions on the role determine which fields are available. For example, an administrator or supervisor can complete system management tasks such as creating users and groups, modifying user permissions, and importing information.



To learn more about system administration tasks, see “PolicyCenter Administration” on page 633.

QuickJump

You can use the **QuickJump** text box in PolicyCenter to enter navigation commands. The **QuickJump** box allows you to quickly jump to other parts of the application.

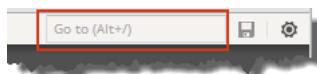
This topic includes:

- “QuickJump Overview” on page 51
- “Using QuickJump” on page 52
- “Configuring QuickJump” on page 52
- “QuickJump References” on page 53

QuickJump Overview

The **QuickJump** box provides a fast way for you to navigate elsewhere in the application. The box also provides auto-complete for both commands and parameters. Auto-complete means that you can type the first few letters of your destination, and the **QuickJump** box provides a list of any matching destinations. For example, typing the letter A displays all destinations that begin with A. Next, select your destination.

This configurable feature appears at the upper right corner of each PolicyCenter screen.



Jump to a Screen or Entity

You can jump to particular screens (absolute commands) or entities. See “QuickJump References” on page 53 for the list of commands that PolicyCenter provides.

Some of the **Screens** are the:

- Desktop tab
- Search tab
- Team tab

- Admin tab

Entities that PolicyCenter provides are **Policy** and **Account**; however, you can add others. To reach a specific destination you can append an argument (only one) or add another related entity.

For example, to retrieve an account, type **account** and the account number, *Account C000143542*, and you jump to the **Account File Summary** screen. If you want to see a policy, type **policy** and the policy number, *Policy 25-123436*, and you jump to the **Policy File Summary** screen.

Note: QuickJump is not available in pop-ups.

QuickJump Behavior in Wizards

Usually you use wizards to take you through a series of consecutive steps. However, in certain instances you can use QuickJump within a wizard:

- You can QuickJump to screens that are available on the left sidebar of the user interface.
- QuickJump actions in a wizard are only available when you are in that wizard. For example, you cannot jump from an account screen to a specific screen in a wizard.

Note: If you are in a wizard and decide to jump to another part of PolicyCenter, your work will be lost if you do not save it first.

Using QuickJump

If the QuickJump box contains at least one command, then it appears on every screen in PolicyCenter.

- **Using Only Keyboard Commands** – Type Alt /. The QuickJump box displays a list of commands that you can select.
- **Using the QuickJump Box in the User Interface** – Place your cursor in the QuickJump box and a list of commands appears.

Chaining QuickJump Destinations Together

You can jump to a specific destination, such as an account's policy. To chain QuickJump destinations, enter the entity name, add a space, then add the second entity name. For example, while viewing an account on the **Account** tab, suppose you want to retrieve information on a policy transaction. You can type **Account C000212105**, type a space and the letter A. The screen displays a list of commands beginning with the letter A that are relevant to the account. Selecting **Account C00021215 AccountWorkOrders** takes you to that screen.

Configuring QuickJump

You can configure the QuickJump box in the following ways:

- Remove QuickJump from the user interface.
- Add commands to jump to newly created screens.
- Change an existing command. Make this change if users have memorized commands on previous systems and want to keep them.
- Add an absolute jump command.

Studio Contains a QuickJump Configuration Editor

You can find the QuickJump resources in the QuickJump configuration tab under **Page Configuration (PCF)** in Studio. Selecting this resource opens the editor, which allows you to add, remove and rename QuickJump commands. For more information, see “[Implementing QuickJump Commands](#)” on page 113 in the *Configuration Guide*.

QuickJump References

The following tables list the QuickJump items that PolicyCenter provides. Some commands can be chained, that is to say, appended with other information, such as another entity name or a policy number.

Static Items

Screen	Command
Account → Account File Summary	Enter Account followed immediately by the account number (no space)
Account → New Submissions	NewSubmission
Administration tab	Admin
Administration → Business Settings → Activity Patterns	ActivitySearch
Administration → Users & Security → Organizations	OrganizationSearch
Administration → Users & Security → Producer Code Search	AdminProducerCodeSearch
Administration → Users & Security → Producer Codes	AdminProducerCodeSearch
Administration → Users & Security → Users	UserSearch
Desktop tab	Desktop
Desktop → My Activities	MyActivities
Desktop → My Other Policy Transactions	MyOtherTransactions
Desktop → My queues	MyQueues
Desktop → My Renewals	MyRenewals
Desktop → My Submissions	MySubmissions
Policy → Summary	Enter Policy followed immediately by the policy number (no space)
Search → Search Accounts	AccountSearch
Search → Search Policies	PolicySearch
Search → Search Producer Codes	ProducerCodeSearch

Super User Items

Screen	Command
Administration → Business Settings → Activity Patterns	Activity Patterns
Administration → Monitoring → Event Messages	Event Messages
Administration → Monitoring → Workflows	Workflows
Administration → Users & Security → Attributes	Attributes
Administration → Users & Security → Regions	Regions
Administration → Users & Security → Roles	Roles
Administration → Utilities → Script Parameters	ScriptParameters
Team tab	Team
Team → Other Policy Transactions	TeamPolicyChanges
Team → Team Activities	TeamActivities
Team → Team Renewals	TeamRenewals
Team → Team Submissions	TeamSubmissions
Team → Team Summary	TeamSummary

Account File Items

The following items are accessible in the **QuickJump** box from within an account file, policy file, or policy transactions that contain an account object. They can also be chained to the end of the **Account** command.

Screen	Command
Account Roles	Roles
Billing	Billing
Claims	Claims
Account File Contacts	Contacts
Account File Documents	Documents
Account File Locations	Locations
New Document – Create a new document from a template	NewDocumentCreate
New Document – Link to an existing document on this account	NewDocumentLink
New Note	NewNotes
Account File Notes	Notes
Account File Contacts → Role	Roles
Submission Manager	SubmissionManager
Account File Summary	Summary
Underwriting File	UnderwritingFiles

Policy File Items

The following items are accessible with the **QuickJump** box only from within a policy file. They can also be chained to the end of the **Policy** command.

Screen	Command
Audit Schedule	Audit
Billing	Billing
Cancel Policy	CancelPolicy
Change Policy	ChangePolicy
Contacts	Contacts
Documents	Documents
History	History
Locations	Locations
New Document – Create a new document from a template	NewDocumentCreate
New Document – Link an existing document to this policy	NewDocumentLink
New Note	PolicyNewNotes
Notes	Notes
Participants	Participants
Payment	Payment
Policy Info	PolicyInfo
Pre-Renewal Direction	PreRenewal
Quote	PolicyQuote
Referral Reason	Referral
Risk Analysis	RiskAnalysis
Summary	Summary



chapter 4

Basic Search

PolicyCenter provides basic search which is a free-text search for quick access against very large databases. Free-text search also provides exact and inexact matching. Inexact matching returns results that partially match, are synonyms, and sound-like the search criteria.

This topic includes:

- “Basic Search Overview” on page 55
- “Basic Policy Search Overview” on page 57
- “Basic Search User Interface” on page 57
- “Working with Basic Search” on page 60

Note: In addition to basic search, PolicyCenter also includes advanced search. Advanced search uses database search, which directly searches the PolicyCenter database. PolicyCenter provides advanced search for policies, policy transactions, accounts, producer codes, activities, and contacts. For more information, see “Advanced Search” on page 63.

Basic Search Overview

In PolicyCenter, basic search uses free-text search which provides faster search than database search against very large databases. The search is faster because it searches through text-based representations of selected data. You can choose to enable or disable basic search. Basic search is disabled in the default configuration.

PolicyCenter includes basic search for policies and submissions. You can configure basic search for policies and submissions.

The **Search Policies → Basic** screen has fields to enter data by name, address, and other criteria. For each field, there is a corresponding search index to optimize retrieval of that data. One search field may map to more than one object or property in the database. For example, entering a value in the **Name** field compares the search string against an index field that consists of concatenated **First Name** and **Last Name** or **Company Name**.

In PolicyCenter, free-text search uses an integration with the full-text search engine Solr. Free-text search is disabled by default. For more information on enabling and configuring free-text search, see “Free-text Search Configuration” on page 339 in the *Configuration Guide*.

Indexing Free-text Search Data

The free-text search process consists of three steps:

1. Initial population of the search index database using a batch process.
2. Continuous index updates in production using messaging.
3. Executing a search query to Solr server.

As users make and save changes, PolicyCenter updates the indexes dynamically.

Query and Filter Search Fields

On the search screen, the fields for entering search criteria are of two types:

- Query fields – PolicyCenter sends the query fields to the search engine.
- Filter fields – Narrow down the results returned by the query.

Basic search requires that you enter at least one query field. PolicyCenter displays a message if you do not specify at least one query field.

Exact or Inexact Search and Ranking

Search fields are configured to match exactly or inexactly. An exact match of a field returns a result that matches the search string exactly. An inexact match of a field returns a result that starts with, contains, is a synonym of, or sounds like the search string.

For example, exact and inexact matching returns the following names if you search for Mary:

- Mary – Exact.
- Marybeth – Starts with.
- Rosemary – Contains.
- Molly – Synonym. A synonym is a word that has the same meaning. For names, you can think of a synonym as a nickname.
- Marie – Sounds like.

PolicyCenter ranks the search results with a score that reflects the degree to which the result matches the search criteria.

A configuration file defines for each search field how to rank exact matches and the various types of inexact matches. For more information, see “Configuring Free-text Search for Indexing and Searching” on page 343 in the *Configuration Guide*.

Search Results for Fields that Match Two or More Similar Pieces of Information

A search field may return matches from two or more pieces of information on the search object. The search ranks the matching information.

When searching for policies for example, a name search attempts to match the names of the primary named insured and additional named insureds on the policy. However, the results display only the **Name of the Primary Named Insured**.

Assume you have a policy that insures Ray Newton as the primary named insured. Christina Newton and Maggie Newton are additional named insureds. You search for Maggie Newton. The search finds Maggie on the Ray Newton policy and displays the Ray Newton’s name on the policy.

Although you searched for Maggie Newton, the **Name** field in the results displays Ray Newton. The  symbol appears after Ray Newton. Hover over the symbol to view Maggie Newton as an additional named insured.

Basic Policy Search Overview

In the default configuration, PolicyCenter contains basic search for policies. Basic search for policies has the following features:

- The **Name** search field matches a concatenated first and last name or company name.
- The **Name** search field matches the primary named insured or additional named insured on the policy. In personal auto line of business, the search also matches the secondary named insured.
- The **Phone** search field matches a home, mobile, fax or work phone number on the policy.
- The **Address** fields match against the current and former addresses on the policy. If there are multiple addresses, the **Address** field in the search results only displays the matching address.
- Basic search returns bound policies and unbound policy transactions.
- Basic search does not return archived policies.

Note: PolicyCenter includes advanced search for additional object types. For more information, see “Advanced Search” on page 63.

Basic Search User Interface

This topic describes the basic search user interface.

Policy Basic Search Screen

To access policy basic search, navigate to **Search → Policies** and view the **Basic** tab. Enter search criteria in the top of this screen, and PolicyCenter displays results at the bottom.

Basic Search Criteria for Policies

On the **Basic** tab of the **Search Policies** screen, the following search fields appear at the top of the screen.

Field	Description	Matching	Type
Policy Number	Search for a policy number. This field requires an exact match or a match that contains the search string. A result that starts with the search string has better search score than a string that only contains the search string.	Inexact	Query
Name	Search for first and last name of a person or company name. Searches for matches in primary named insured and additional named insureds. For details, see “Basic Name Search” on page 59.	Inexact	Query
Phone	Search for a matching work, home, mobile, or fax phone number. You must enter the whole phone number. Valid telephone number formats are: <ul style="list-style-type: none">• 650-555-1234• 650 555 1234• 6505551234• (650)555-1234• (650) 555-1234• 650.555.1234	Exact	Query
Official ID	Search for a Social Security number (SSN) or employer identification number (EIN) number.	Exact	Query

Field	Description	Matching	Type
Address			
Street	Search for the street address. The search ranks the results from highest to lowest as follows: <ul style="list-style-type: none">• Exact• Starts with• Sounds like• Contains	Inexact	Query
City	Search for the city. The search ranks the results from highest to lowest as follows: <ul style="list-style-type: none">• Exact• Starts with• Sounds like• Contains	Inexact	Query
State	Search for the state.	Exact	Filter
Postal Code	Search for the postal code.	Exact	Filter
Filters			
Product	Search for the product of the policy or policy transaction.	Exact	Filter
Jurisdiction	Search for the jurisdiction of the policy or policy transaction.	Exact	Filter
Producer of Record	Search for policies or policy transactions owned by a particular producer of record.	Exact	Filter
Producer Code	Search for the producer code of service for the policy or policy transaction.	Exact	Filter
In Force On	Search for policies or policy transactions in force on this date.	Exact	Filter

The Matching column indicates whether the field matches exactly or inexactly. For more information, see “Exact or Inexact Search and Ranking” on page 56.

The Filter column indicates whether the field is a query or filter field. You must specify at least one query field such as **Policy Number** or **Name**. For more information, see “Query and Filter Search Fields” on page 56.

Basic Search Results for Policies

On the **Basic** tab of the **Search Policies** screen, the following **Search Results** fields appear at the bottom of the screen.

Field	Description
Result type	Displays an icon representing the result type. The result types are: <ul style="list-style-type: none">• – The policy icon represents a bound policy.• – The policy transaction icon represents a policy transaction, such as a submission or policy change.
Rank	The rank indicates the relevance of the result to the search criteria. The lowest rank corresponds to the most relevant match.
Policy #	The policy number. If the result is not a bound policy period and does not have a policy number, Unassigned appears in this column.
Name	The first and last name of the person or the company name returned by the search results. This field displays the primary named insured on the policy. The symbol appears after the name if there are additional named insureds on the policy. Hover over the symbol to view the names of the additional named insureds.
Address	The policy address of the policy.
Product	The product of the policy or policy transaction.

Field	Description
Status	The status of the policy or policy transaction.
Effective Date	The effective date for the policy term.
Expiration Date	The expiration date for the policy term.
Producer	The Organization and Producer Code as it appears in the Producer of Service on the Policy Info screen.

Basic Name Search

The **Name** field finds matches in the primary named insured and additional names insureds on a policy. A match on primary named insured has a better ranking than additional named insureds. This is an inexact search field.

Starting with the best match, basic search ranks the matching names as follows:

1. Exact
2. Starts with
3. Synonym
4. Sounds-like
5. Contains

If you enter more than one word in the name field, the search gives a better rank to results containing both words. A match has a better ranking if the words exist in the same order. If only part of the words match, the match has an inferior ranking.

A multiple word search that fully matches the additional named insured has a better rank than a match that only partially matches the primary named insured. For example, you search for Ray Newton. Policy 1 has Ray Newton as an additional named insured, and policy 2 has Ray Brussard as a primary named insured. Policy 1 has a better ranking than policy 2.

You can search for historical names, such as maiden names, on a policy over time. For more information, see “Basic In Force On Search” on page 59.

Basic Address Search

The address search finds current and historical addresses. Search returns a matching address in the primary address of a primary named insured or additional named insured. Search returns a result if an address matches the Street and/or City fields. These fields are query fields. Basic search filters the query results by State and Postal Code fields. These fields are filter fields.

Basic In Force On Search

The **In Force On** filter field returns only those policies that are in force on that date.

For example, Robert Brown took out policy on January 1, 2002. He renews the policy every year but makes no other changes. To view the policy in force on October 1, 2004, set the **In Force On** field to this date. The search returns the policy period in force on this date.

The **In Force On** search has special behavior for the **Name** field. Suppose you enter an **In Force On** date and specify the **Name** field. The search returns only policies that were in force on that date and have that name at some point on the policy. For example, Jane Doe has a policy in force from January 1, 2012 through January 1, 2013. Jane married and changed her name to Jane Smith on July 1, 2012. You search for either Jane Doe or Jane Smith with an in force date of August 1, 2012. The search returns Jane’s policy that is in force on August 1. If you search for Jane Smith with an in force date of August 1, 2011, then the search returns no matches.

Working with Basic Search

This topic provides step-by-step instructions for working with basic search.

Prerequisites

These examples assume that you have loaded the **Free-text Search** sample data set. For more information about loading sample data, see “[Installing Sample Data](#)” on page 52 in the *Installation Guide*.

Producer Code Access Controls Basic Search Results

Basic search results returns policies for which the user has producer code access. If producer code security is enabled, the producer of service on the policy must match one of the user’s producer codes. For more information, see “[Producer Code Security and Policies](#)” on page 639 and “[Turning On Producer Code Security](#)” on page 648.

Basic Policy Search Examples

This topic provides step-by-step instructions for doing some simple policy searches.

1. Select **Search** → **Policies** to navigate to the **Search Policies** → **Basic** tab.

2. In **Name**, enter **ray**, then click **Search**.

The **Search Results** displays policies which contain a primary or additional named insured with **ray** in the first name or last name. For example, the results contain rows for Ray Newton and Ann-Marie Ray.

In the **Name** field, a number of entries have the  symbol at the end.

3. Hover over the symbol next to **Ann-Marie Ray**.

PolicyCenter displays **Ray's Rockhouse** as an additional named insureds on the policy.

4. In **Name**, enter **rock**, then click **Search**.

PolicyCenter displays Ann-Marie’s policy in the results because the search string matches **Ray's Rockhouse**, an additional named insured on the policy.

5. Click the **Policy #** link to jump to the policy.

If the search result is a bound policy, then PolicyCenter opens the policy file.

If the result is an unbound policy period, then PolicyCenter opens the policy transaction (job) wizard for that policy period.

In the result type column, the policy transaction icon  appears for unbound policy periods such as a submission or other policy transaction. The policy icon  appears for bound policy periods.

6. Click **Policy Info** in the left sidebar.

Notice that **Additional Named Insureds** includes **Ray's Rockhouse**.

Search Index Updates Automatically

In PolicyCenter, if you commit a change that is part of the search index, that change is updated automatically in the search index database.

These instructions continue “[Basic Policy Search Examples](#)” on page 60.

1. Go to the Ann-Marie Ray policy.
2. Select **Actions** → **Change Policy**.
3. Advance to the **Policy Info** screen.

4. On the **Policy Info** screen, change the primary named insured to **John Smith**.
5. Click **Quote** or **Save Draft**.
6. Copy the policy number, then return to the **Search Policies → Basic** screen.
7. Enter the **Policy Number** and click **Search**.

The search results display the John Smith policy as a policy transaction and the Ann-Marie Ray policy as a bound policy.

You may have to wait a short time for the index update to occur.



chapter 5

Advanced Search

PolicyCenter includes advanced search to find matching policies, policy transactions, accounts, producer codes, activities, and contacts. The advanced search uses a database search of the PolicyCenter database. This topic describes advanced search in PolicyCenter.

This topic includes:

- “Advanced Search Overview” on page 63
- “Working with the Advanced Search Tab” on page 65

Note: In addition to advanced search, PolicyCenter also includes basic search that uses free-text search. Free-text search improves performance for large data sets and includes inexact matching. PolicyCenter provides basic search for policies. For more information, see “Basic Search” on page 55.

Advanced Search Overview

You can access the advanced search from the **Search** tab. For policies, you access advanced search on the **Advanced** screen in the **Search → Policies**.

Note: The **Search Policies → Basic** screen uses free-text search not database search. For more information, see “Basic Search” on page 55.

Some fields on the advanced search screens are text fields. If you enter text into one of these fields, PolicyCenter searches for a match that starts with that text. For example, if you enter *Jones* into the last name field, the search returns all last names that start with *Jones*. The search results include: *Jones*, *Jonesburg*, or *Jones-Smith*. It does not find *McJones*.

You must enter an exact match in the **Account Number** and **Policy Number** fields.

During a search, PolicyCenter uses only those fields in the form in which data exists. For example, if you search for a **Policy** and enter a **Last Name** but not a **Policy Number**, PolicyCenter omits **Policy Number** from the search.

The search results returns accounts, policies, or policy transactions with links to view details. Accounts, policies, or policy transactions for which you do not have sufficient producer code permissions do not appear in the search results.

See also

- “Data-based Security for Accounts and Policies” on page 638
- “Configuring Search Functionality” on page 329 in the *Configuration Guide*

Minimum Search Requirements

This topic describes the minimal search requirements on the following search screens:

- **Search Policies → Advanced**
- **Search Accounts**
- **Search Contacts**

You can search on the following fields without specifying additional search information:

- **Official ID**
- **Account Number**
- **Policy Number**
- **Phone**

Personal Names

Both the first and last name fields have a check box to indicate whether the name must be an exact match.

The name of the person requires the following:

- Both first and last name.
- For the first or last name, the name must be an exact match. If exact match is not selected, you must provide the first three letters of the name.
- If the last name is not an exact match, you must provide either city and state or postal code.

Company Names

There is now a check box to specify whether the company name is an exact match.

The name of the company requires the following:

- The name must be an exact match, or you must provide the first five letters of the name.

Producer Code

You can enter a producer code without specifying additional search information.

Phone Number

On the search screen, the **Phone** field matches the contact's **Work Phone**. Searches on phone number require an exact match. The extension for a phone number is stored in the same field as the phone number itself. If a phone number has an extension, a search omitting the extension is not a match.

The phone number matches the contact's work phone number.

Advanced Search for Policies

The **Search Policies → Advanced** screen allows you to search for policies and policy transactions. By using the **Search For** drop-down menu, you can search for the following policy transaction types:

- **Cancellation**
- **Final Audit**
- **Policy Change**
- **Premium Report**
- **Reinstatement**
- **Renewal**
- **Rewrite**
- **Submission**

When you choose **Final Audit** or **Premium Report**, options appear that allow you to search by date. You can search by **Audit period end date** or **Audit due date** and specify a date range. This search finds already started audit policy transactions. The search does not find audit policy transactions with a status of **Scheduled**. Managers can use this search to find all final audits due between a set of dates. Then the manager can assign the audits.

In the **Search Policies → Advanced** screen, you can assign a user to one or more policies or open policy transactions. For the selected policies or policy transactions, you can choose a user for an assignment role such as auditor, producer, or underwriter. The selected user replaces the user who previously held that assignment role. If a user is a member of more than one group, you must also assign the group. When you assign the user, the group is also assigned. You cannot assign a user to completed or bound policy transaction.

See also

- “Working with the Advanced Search Tab” on page 65
- “Assigning Submissions, Renewals, and Other Policy Transactions” on page 662 for a similar feature on the **Team** tab.

Search Contacts

The **Search → Contacts** menu item displays the same screen as the **Contact → Search** menu item. For more information, see “Searching for a Contact on the Contact Tab” on page 375.

Click the contact **Name** to display the **Contact File Details** screen. For more information about the **Contact File Details** screen, see “Contact Tab” on page 47.

Working with the Advanced Search Tab

In the **Search Policies → Advanced** screen, you can assign a user to one or more policies or open policy transactions. For the selected policies or policy transactions, you can choose a user for an assignment role such as auditor, producer, or underwriter. The selected user replaces the user who previously held that assignment role. If a user is a member of more than one group, you must also assign the group. When you assign the user, the group is also assigned. You cannot assign a user to completed or bound policy transaction.

Assigning a user to one or more policies or open policy transactions

1. Select **Search → Policies** and click the **Advanced** tab.
2. Make a selection from the **Search For** drop-down list. For example, you can select **Policy** or a policy transaction such as **Submission**.
3. Enter search criteria and click **Search**. You must meet the minimal search criteria as described in “Minimum Search Requirements” on page 64.
If you have loaded the small sample data set, the following searches return results:
 - Enter the **Producer Code** as 100-002541.
 - Enter **First Name** as Ray and **Last Name** as Newton.
4. Select one or more policies or policy transactions and select an assignment role from the **Assign** drop-down list.
Note: You cannot assign a user to completed or bound policy transactions. Therefore, the **Assign** button is disabled if the selection includes one of these policy transactions.
PolicyCenter displays the **Assign transactions** screen. For each policy or policy transaction, this screen displays the type, policy transaction or policy number, and assignment role.
5. Enter a **User Name**, **First Name**, or **Last Name** and click **Search**.

PolicyCenter displays matching users. For each user, PolicyCenter displays the **Group** and **Parent Group**. A user appears multiple times if the user belongs to more than one group. For example, you can search for users with last name Applegate. The search returns two rows for Alice Applegate because Alice is a member of the Eastern Region Underwriting and Los Angeles Branch UW groups.

6. In the search results, click **Assign** to assign a user for the chosen role. If a user is a member of more than one group, select the user's row that displays the group of your choice.

See also

- “[Assigning Submissions, Renewals, and Other Policy Transactions](#)” on page 662 for a similar feature on the **Team** tab.

Saving Your Work

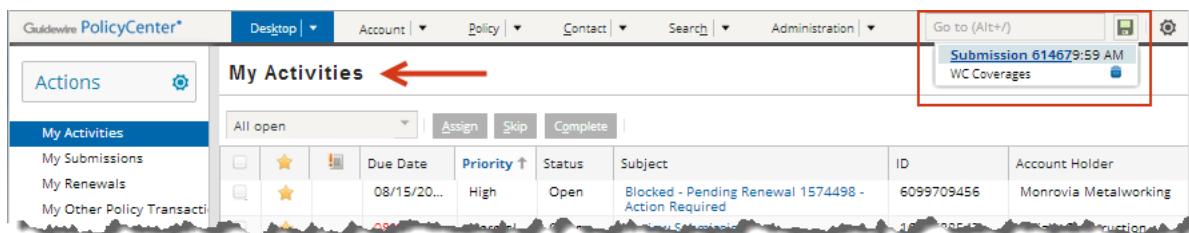
PolicyCenter automatically saves your work to the database in wizards and through the **Unsaved Work** menu in the user interface.

This topic includes:

- “[Unsaved Work Menu](#)” on page 67
- “[Saving Your Work in PolicyCenter Wizards](#)” on page 68

Unsaved Work Menu

You can access your unsaved work from the **Unsaved Work** menu in the Tab Bar. PolicyCenter automatically saves your work whether you are in the **Account** or **Administration** tabs or in a wizard such as a submission or policy change. If you leave one of those screens with unsaved changes, and navigate to another section of PolicyCenter, the server keeps your information in memory. The database does not store your information. You are able to retrieve your work from the **Unsaved Work** menu, as seen in the following example. Selecting it returns you to that screen with your unsaved data. You can finish your work and save it by clicking **Update**.



The **Unsaved Work** menu is useful if you must navigate away from a screen but need to return to it later. After you complete and save your work, PolicyCenter removes that item from the **Unsaved Work** menu. However, if you attempt to log out without saving, PolicyCenter alerts you that your unsaved work will be lost if you continue.

Autosaving is the mechanism PolicyCenter uses to save work that can be retrieved by using the **Unsaved Work** menu.

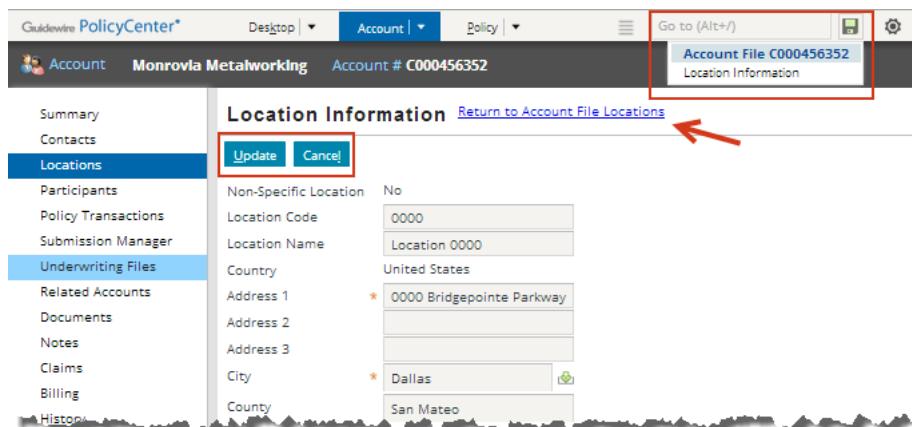
Both `autosaveable` and `countsAsWork` default to `true` in the default application, which you can see in the PCF Editor in Studio.

Saving Your Work in PolicyCenter Wizards

When you work in any policy transaction (job) wizard, PolicyCenter saves your information to the database. It saves every time you advance or go back in the wizard (by clicking wizard buttons such as **Back**, **Next**, **Quote**, or **Bind**). It also saves when you navigate to a different step by using the sidebar menu choices. However, there are times when you are in a PolicyCenter wizard, where you may need to enter additional information that is technically not in a wizard step. Popup windows are an example of this.

Popup Window Behavior

In PolicyCenter, a popup is a PCF element that can be edited in Studio. It is part of the wizard process, but it is not a wizard step. In the following example, **Location Information** is a popup.



When you make changes or add information in a popup, PolicyCenter does not commit those changes immediately to the database, but saves the changes to the enclosing parent page. The **Unsaved Work** menu captures the changes to the parent page and the popup so that you can finish your work if you navigate away without saving the parent page. For example, suppose you modify a location in a workers' compensation submission and click **OK**. Then you navigate to the **Desktop**, or log out without first clicking **Save Draft**. In this case, the new location data is not saved to the database, but will be stored in the **Unsaved Work** menu.

Localizing PolicyCenter

You can localize your PolicyCenter interface so that it displays information in a language other than US English. If PolicyCenter has multiple locales defined, a language selector appears along the top of the PolicyCenter interface.

You can change the date, time, number, and currency formats for values that appear in PolicyCenter. You can also localize the product model.

This topic includes:

- “Localizing Documents, Email, and Notes” on page 69
- “Localizing Activities” on page 69
- “Localizing Workflows” on page 70

Localizing Documents, Email, and Notes

A system configurator can create localized versions of document, email, and note templates. Then, in PolicyCenter, a user of PolicyCenter can use these templates to create a document, note, or email in the language of their locale.

Localizing Activities

A PolicyCenter administrator can localize the subject and description of an activity pattern from the Administration tab. Then, within PolicyCenter, a user can see the subject and description of the activity in the language of their locale.

See also

- “Localizing Administration Data” on page 45 in the *Globalization Guide*

Localizing Workflows

Before advancing a workflow, PolicyCenter determines the locale in which to execute the workflow. PolicyCenter uses this locale for display keys, dates, numbers, and other similar items generated during the execution of the workflow. At the start of the workflow execution, PolicyCenter evaluates the workflow locale and uses that locale for notes, documents, activities, and similar items. PolicyCenter workflows are executed in the preferred language of the policy being processed. In PolicyCenter, an administrator can also see localized versions of workflow logs and workflow step names in Studio.

See also

- “Localizing Guidewire Workflow” on page 49 in the *Globalization Guide*

part III

PolicyCenter Policy Transactions

Policy Transactions

On a daily basis, producers and agents do work associated with policies. This work includes creating submissions, changing policies mid-term, and any number of similar activities. In PolicyCenter, you do this work in *policy transactions*. Policy transactions play a central role in PolicyCenter. This topic provides an introduction to policy transactions and describes how policy transactions process information. Subsequent topics contain details on each type of policy transaction.

This topic includes:

- “Overview of Policy Transactions” on page 73
- “Key Features of Policy Transactions” on page 76
- “Configuring Policy Transactions” on page 78

See also

- “Policy File” on page 303

Overview of Policy Transactions

Policy transactions coordinate all the work associated with creating a new policy period and modifying the policy. Policy transactions are almost always referred to by type, that is to say, a submission, a policy change, or a cancellation.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

PolicyCenter has the following types of policy transactions:

- “Submission” on page 74
- “Issuance” on page 74
- “Renewal” on page 74

- “Cancellation” on page 74
- “Policy Change” on page 75
- “Reinstatement” on page 75
- “Rewrite” on page 75
- “Rewrite New Account” on page 75
- “Audit” on page 75

Submission

Submission is the only policy transaction that creates a policy. A potential policyholder contacts the carrier or agent and requests a quote. The agent gathers information in order to generate one or more quotes. Based upon the apparent risk of policyholder, PolicyCenter raises underwriting issues that may require approval. If both parties agree upon a quote, then the agent binds and (optionally) issues the policy.

See also

- “Submission Policy Transaction” on page 79

Issuance

Issuance is part of the submission process. It allows you to edit and requote a bound submission before officially issuing the policy (sending out the accompanying policy forms). For example, a potential customer has a new limousine business and must insure all 30 vehicles today. The customer contacts you, the insurance agent, requesting a business auto policy. You require the VIN number and license of all vehicles, but the customer does not have these readily available. You still proceed with generating a quote and agreeing on the terms. The policy is bound (legal) today, so the customer’s limousines have coverage. The next day, the customer contacts you, provides the required information, and adds another limousine, bringing the total number of vehicles to 31. You edit, requote, and now issue the policy by using an issuance policy transaction.

See also

- “Issuance Policy Transaction” on page 89

Renewal

The renewal process extends the policy for another term beyond the current expiration date. It creates a new policy period for an existing policy.

The renewal policy transaction is often automatic. For example, if there are no changes to the policy and no claims were made against it, the system creates a new policy period and sends a renewal notice. Renewal can also require that an underwriter review the policy. Processing occurs prior to expiration, but actual renewal is at expiration. Like submissions, you can create one or more quotes on a renewal.

See also

- “Renewal Policy Transaction” on page 93

Cancellation

The cancellation process is a type of policy change which marks a policy as canceled. A cancellation can be initiated by the carrier. A cancellation initiated by the carrier typically requires advance notice to the policyholder. Therefore, the carrier starts the cancellation on one date, and the cancellation completes some period of time later. For example, a policyholder forgets to pay his auto policy by the due date of June 10th. On June 11th, the system starts a cancellation policy transaction for non-payment with termination of coverage effective as of a future date. The future date is usually based on regulatory requirements.

A policyholder can also initiate a cancellation. For any number of reasons, a policyholder may no longer want coverage by the carrier. According to the policyholder's wishes, the carrier cancels the policy effective immediately or at some future date.

See also

- “Cancellation Policy Transaction” on page 107

Policy Change

To create a policy change, you modify a policy in between the effective and expiration dates. A change can be as simple as adding an additional vehicle to an auto policy. Or it can be an out-of-sequence event, such as adding another driver to a policy on a date prior to the addition of another vehicle to the policy.

See also

- “Policy Change Transaction” on page 115

Reinstatement

Reinstatements go hand in hand with cancellations and are a type of policy change that *uncancels* the policy. Reinstate a canceled policy. The reinstatement date must be the same as the cancellation effective date.

See also

- “Reinstatement Policy Transactions” on page 125

Rewrite

Policies are rewritten to make the types of changes that cannot be done in a policy change policy transaction, to correct significant errors, or to make changes to the policy. A rewrite, which can only occur on a canceled policy, effectively ends the first policy and creates a new one in its place. For example, a customer requests a workers' compensation policy. However, when the customer receives the policy, he notices many errors: the dates and payroll amounts are incorrect, and the building and location are in the wrong jurisdiction. The customer notifies you, the agent. If you choose to fix the errors in a policy change, the system would send out an addendum, calling out the mistakes in the policy. But because there are so many mistakes in the policy, you decide to rewrite the policy which sends out completely new policy documentation.

See also

- “Rewrite Policy Transactions” on page 129

Rewrite New Account

When you rewrite a policy to a new account, PolicyCenter creates a rewrite new account policy transaction. This policy transaction takes data from an existing policy and creates a new policy with a new policy number in the new account. Unlike a rewrite policy transaction, a rewrite new policy transaction can have pre-qualification questions. You can only rewrite canceled or expired policies to a new account.

See also

- “Rewrite New Account Policy Transaction” on page 135

Audit

The audit policy transaction lets the carrier verify information about the policyholder and determine the accuracy of premiums paid. The audit policy transaction provides final audit and premium reports.

PolicyCenter supports final audit for the workers' compensation line of business. You set up the method of final audit (physical, voluntary, or by phone) when you create the workers' compensation policy. PolicyCenter creates audits when the current time reaches the initiation date of an audit schedule item. Unlike other policy transactions, the audit policy transaction does not create a new version of the policy, and therefore does not affect the coverage.

With premium reports the policyholder is billed for premium based on periodic requests for actual basis amounts, such as payroll. A deposit, usually a percentage of the estimated annual premium, is billed at the beginning of the policy. As each reporting period ends, the policyholder is billed based on the actual basis reported by them.

See also

- “Premium Audit Policy Transaction” on page 143

Key Features of Policy Transactions

There are a number of policy transaction features that apply to one or more policy transaction types. In the following table, the marked cells indicate which features are available for each policy transaction type in the default application.

Common Feature	Submission	Issuance	Policy Change	Cancellation	Reinstatement	Rewrite	Rewrite New Account	Renewal	Audit
Can get new policy number	●					●	●	●	●
Change policy effective date	●	●				●	●	●	●
Change policy expiration date	●	●	●			●	●	●	●
Change producer code of record	●	●				●	●	●	●
Change producer code of service	●	●	●			●	●	●	●
Create new period	●					●	●	●	●
Decline	●		●						●
Multiple versions	●		●						●
Not taken or non-renewed	●								●
Policy holds	●	●	●		●	●	●	●	●
Qualification questions	●						●		
Quick quote	●								
Referral reasons copied over to underwriting issue	●	●	●		●	●	●	●	●
Select UW company	●	●				●	●	●	●
Underwriting issues block progress	●	●	●		●	●	●	●	●
UW approval	●	●	●		●	●	●	●	●

The following features are common to various policy transactions:

- “Preemption” on page 77
- “Out-of-sequence Changes and Policy Transactions” on page 77

See also

- “Underwriting Authority” on page 411
- “Policy Holds Administration” on page 665

Preemption

Preemption occurs if there are two or more concurrent policy transactions in process on a policy. When one policy transaction finishes, the other policy transactions needs to adjust to the fact that the policy information has changed. The first policy transaction that finishes preempts any other changes in concurrent policy transactions.

See also

- “Preempted Jobs” on page 503

Out-of-sequence Changes and Policy Transactions

PolicyCenter supports out-of-sequence policy transactions. A policy transaction is out-of-sequence if its effective date is earlier than the effective date of another policy transaction that is already bound on the policy for that contractual period. PolicyCenter warns you that the policy transaction is out-of-sequence and prompts you to address any conflicts that occur as a result.

When a policy transaction begins, PolicyCenter checks to see if the transaction is out-of-sequence. The transaction is out-of-sequence if at least one completed transaction on the policy has an edit effective date later than the edit effective date of the current policy transaction. So when you click **Start** either on the **Start Policy Change**, **Start Cancellation**, or **Start Reinstatement** screens, PolicyCenter displays a confirmation message about the out of sequence change. If you click **OK**, then the policy transaction starts.

See also

- “Out-of-sequence Jobs” on page 501

Out-of-Sequence Policy Transaction Combinations

Since there are many different types of policy transactions, it is helpful to know which policy transactions may be out-of-sequence with respect to another. The following table shows all possible pairs of policy transactions. A cell marked in row *X* and column *Y* indicates that policy transaction *X* may be out-of-sequence with respect to policy transaction *Y*. Policy transaction *X* may be started and completed even if its edit effective date is earlier than the edit effective date of a previously bound policy transaction *Y*.

Out-of-Sequence Policy Transaction Type	Submission	Issuance	Policy Change	Cancellation	Reinstatement	Rewrite	Rewrite New Account	Renewal	Audit
Submission									
Issuance									
Policy Change			•	•	•	•	•		•
Cancellation			•	•	•	•	•		•
Reinstatement			•	•	•	•	•		•
Rewrite			•	•	•	•	•		•
Rewrite New Account			•	•	•	•	•		

Out-of-Sequence Policy Transaction Type	Submission	Issuance	Policy Change	Cancellation	Reinstatement	Rewrite	Rewrite New Account	Renewal	Audit
Renewal									
Audit									

Note the following:

- **Cancellation and reinstatement** – When you cancel or reinstate a policy, you cannot make additional changes to that policy. Therefore, if a policy transaction is out-of-sequence with respect to a cancellation or reinstatement policy transaction, there are no conflicts to resolve.
- **Rewrite** – If rewrite is out-of-sequence with respect to renewal, then you receive a warning that the rewrite changes are not applied to the renewal.
- **Submissions, issuances, and rewrite new account** – Can never be future-bound policy transactions.
- **Submissions** – Can never be out-of-sequence because no policy exists until a submission is complete.
- **Renewals** – Can never be out-of-sequence because no other policy transactions with later effective dates can be started until the renewal completes. However, if PolicyCenter promotes a renewal period, then any subsequent policy transaction in the expiring period will always be out-of-sequence with respect to the renewal.
- **Audits** – Since PolicyCenter never promotes audit branches, they are never in an out-of-sequence relationship.
- **Rewrite new account** – If rewrite new account is out-of-sequence to another policy transaction, the out-of-sequence policy transaction occurred on the source policy. (Remember that rewrite new account rewrites a source policy to a new target policy on a different account.) That out-of-sequence policy transaction is a transaction on the source policy but not on the target policy. However, both the source and target policies have a slice representing the out-of-sequence policy transaction.

Note: Only cancellation policy transactions can be started on non-issued policies.

Configuring Policy Transactions

Configuring PolicyCenter policy transactions involves making changes to the product model, Gosu classes and rules, page configuration files (PCF files), wizards, configuration parameters, plugins, and sometimes workflows. Each can be configured based on your business requirements. It is the combination of these elements that enables you to create or modify a policy.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

In contrast to policy transactions, the product model defines the types of policies that you can offer to your customers. Each product can contain one or more lines of business offering various coverages. For more information, see “Product Model Overview” on page 471.

See also

- “Configuring Jobs” on page 549 in the *Configuration Guide*

Submission Policy Transaction

Creating a submission policy transaction is one of the most common activities in PolicyCenter. The goal is to bind and issue the submission which turns it into a policy. In a typical scenario, a producer receives an inquiry for coverage, establishes an account, asks some pre-qualification questions. If the answers are correct, meaning that the risks are reasonable, the producer asks the applicant for additional information. If the answers are not correct, then the policy may be referred to underwriting. If both sides agree to terms and price, then PolicyCenter generates the policy and accompanying documents, and the documents are sent to the applicant. The policy is legally binding to both parties.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

This topic explains what a submission is, how to work with submissions in PolicyCenter, and how you can configure it to meet your business requirements.

This topic includes:

- “Submission Overview” on page 80
- “Submission General Steps” on page 81
- “Submission Key Features” on page 82
- “The Submission Manager” on page 86
- “Working with Submissions” on page 86

See also

- “Configuring Submissions” on page 557 in the *Configuration Guide*

Submission Overview

PolicyCenter handles submissions through the submission wizard. After getting account information, the submission wizard guides you through the process of gathering the required information for the policy. Typically, the first and last wizard steps are the same for all lines of business.

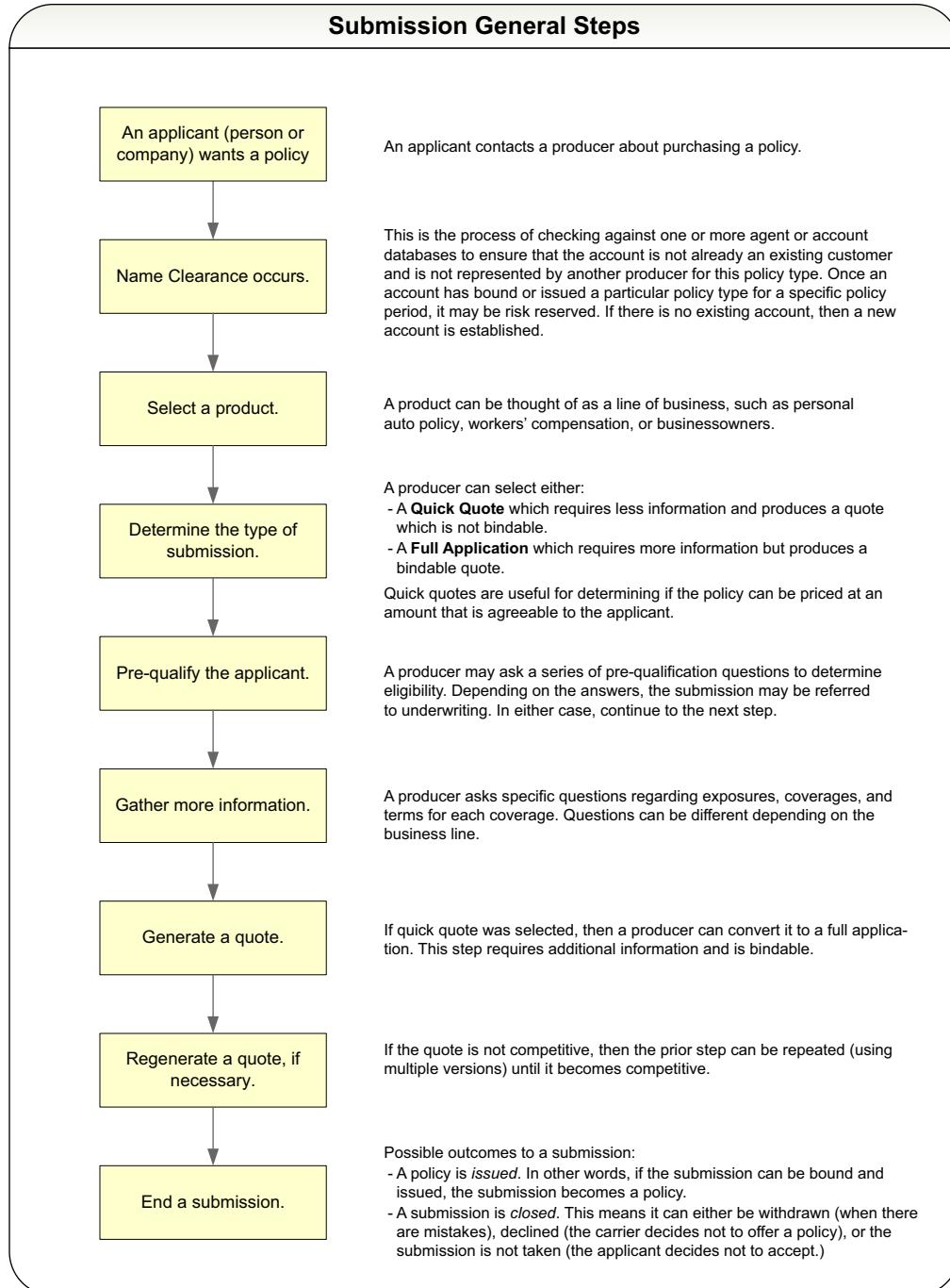
However, the wizard steps differ slightly for each line. In personal auto, some steps gather information about the driver, the type of vehicle, the garage location, and the type of coverages. In workers' compensation, some wizard steps require information about the business, the types of workers, locations, and coverages. So how does a submission end? The outcome goal is to bind and issue the submission. You can choose to issue the policy at a later date. In this case, you search for the policy then run an issuance policy transaction.

See also

- “Account File” on page 323 to learn more about accounts.
- “Issuance Policy Transaction” on page 89 to learn about issuing submissions as a separate policy transaction.

Submission General Steps

The following diagram shows the basic steps to create a submission in the base configuration of PolicyCenter. Your business requirements can alter the process. Also, the steps may differ slightly between personal and commercial lines and in how you access the submission screens.



Submission Key Features

PolicyCenter has feature rich submissions that gives you flexibility in the creation and management of submissions.

- “Understanding Name Clearance and Risk Reservation” on page 82
- “The Difference Between Quick Quote and Full Application” on page 82
- “Answering Pre-qualification Questions” on page 83
- “Selecting an Underwriting Company” on page 83
- “Reasons for Underwriter Review in Submissions” on page 84
- “Creating Multiple Versions” on page 84
- “Closing a Submission” on page 84
- “Differences Between Binding a Policy and Issuing a Policy” on page 85
- “Expiring Submissions” on page 85
- “Copying Submission Information” on page 85

Understanding Name Clearance and Risk Reservation

PolicyCenter has the ability to perform name clearance and risk reservation. It does this through the `IAccountPlugin` methods `performNameClearance` and `isRiskReserved`.

Name Clearance

Name clearance ensures that a person or company is not an existing account and that another producer does not represent them for the given policy type. PolicyCenter checks the name against one or more producer or account databases. You must complete name clearance before creating a new account in PolicyCenter. You can use the `performNameClearance` method to check against external databases when populating the list of available products. This check helps to prevent a carrier from inadvertently competing with itself.

Risk Reservation

In the default application, risk reservation is the process of associating a product and period to a producer code. If a product is risk reserved by a producer code that the current user does not have, then the product’s status on the **New Submission** screen is **Risk reserved**. The current user cannot create new submissions for that product.

The Difference Between Quick Quote and Full Application

In PolicyCenter, you have the option of creating either a quick quote or a full application submission.

- **Quick Quote** gathers the minimal information needed to generate a quote.
- **Full Application** gathers the complete set of information needed to apply for a bindable quote.

What are the Advantages of Quick Quote?

Select **Quick Quote** if you need to generate a quote with minimal information. Quick quote provides a rapid assessment of whether or not the carrier can provide a policy premium that is in the applicant’s price range. If both parties are agreeable to the quote, then additional information must be gathered in a **Full Application** before a policy can be bound. You can generate as many quick quotes as you need and save them, but you cannot bind them. From **Quick Quote**, you can continue to **Full Application**.

In all lines except personal auto, selecting **Quick Quote** skips selected steps in the full submission wizard. Quick quotes skip the **Risk Analysis** and **Policy Review** steps, because PolicyCenter can generate a quote without the infor-

mation requested in these steps. Quick quotes also skip the **Forms** and **Payment** steps, because these items do not apply to a non-bindable quote.

In personal auto submissions, selecting **Quick Quote** does more than skip steps in the full quote submission wizard. Instead, personal auto line quick quotes use a separate wizard that reduces the quote process to two steps, producing quotes for up to two drivers with up to two vehicles. As with any quick quote, the resulting quote is not bindable.

You can continue to create quick quotes if a submission has never had a full application quote generated.

Answering Pre-qualification Questions

Pre-qualification questions are a screening tool used to determine applicant risk and to reflect the carrier's desire for the business. These questions determine if an applicant qualifies for the type of insurance being sought. You access and modify the questions in Guidewire Product Designer. (Not all products in the default application have pre-qualification question sets.) Questions have risk points and correct answers. Depending on the answers, PolicyCenter can raise underwriting issues that cause the policy to be referred to an underwriter. If not approved, these issues may prohibit quoting the policy.

The outcomes of answering the questions are:

- **The applicant answers the questions correctly.** The submission process continues normally.
- **The applicant answers some of the questions incorrectly.** Incorrect answers can raise underwriting issues which an underwriter needs to review. In the base configuration, underwriting issues are raised before quoting the policy.
 - The underwriter reviews the underwriting issues and approves them. The agent can quote the policy and continue the submission.
 - The underwriter decides that the applicant does not meet the base requirements, and declines the underwriting issues. The agent cannot quote the policy. The agent can revise the pre-qualification questions and other parts of the policy. These changes may remove underwriting issues blocking the submission.

You can configure the error message that appears when the applicant answers a question undesirably. In some cases, the carrier wants the message to be as explicit as possible so that agents know exactly what the eligibility requirements are. In other cases, they want the answer to be less straightforward to prevent users from knowing how to beat the system and pre-qualify an undesirable applicant.

Questions can be filtered. They can only be tied to a single answer. For example, if a question has a drop-down list, a filtered question can appear if the user selects a certain value from the drop-down. You cannot configure the system to have a single question appear if the user selects one of two values.

See also

- “Types of Question Sets” on page 60 in the *Product Model Guide*
- “Underwriting Authority” on page 411
- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*

Selecting an Underwriting Company

Large carriers can license more than one underwriting company to underwrite policies on their behalf. The primary reason for having these multiple underwriting companies is to accommodate jurisdictional regulatory requirements. Many jurisdictions do not allow carriers to have more than one set of rates per underwriting company. Therefore, if a carrier wants to offer multiple sets of rates in that jurisdiction, they must file each set of rates under a separate underwriting company for that jurisdiction. Underwriting companies may also offer different coverages or limits.

Underwriters typically profile an account to determine what segment an account falls into, and therefore which underwriting company would actually underwrite the policy. By extension, this determines the set of rates the

account is eligible for. The appropriate set of rates would then be used to quote the policy. For example, a workers' compensation carrier may segment accounts into three different groups: high hazard, medium hazard, and low hazard. The carrier has three underwriting companies that correspond to each segment. To generate a quote for the policy, an underwriter uses the underwriting company automatically selected by PolicyCenter or selects a different underwriting company. The rating engine calculates a quote for the policy based on the underwriting company.

In the **Policy Info** screen, you can select a different underwriting company.

Segmentation can determine which underwriting companies are available for a given submission, since underwriting companies may be able to accept only certain types of risks.

Reasons for Underwriter Review in Submissions

Applications may need to be referred to an underwriter for manual review and approval. Some of the reasons an underwriter might review them are:

- PolicyCenter raised underwriting issues because:
 - Choices on the policy period are considered too risky. For example, the carrier raises an underwriting issue for cars valued at over \$100,000 on a personal auto policy.
 - There are referral reasons on the policy. For example, there have been too many claims in previous policy periods.
 - The default pricing of the policy has been overridden through rating overrides.
 - The applicant did not answer the pre-qualification questions correctly.
- The producer did not have the permission to bind submissions. In this case, the **Bind Options** menu choice triggers the underwriting review.

See also

- “Underwriting Authority” on page 411
- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*

Creating Multiple Versions

You can create multiple versions of a quote to help you and the insured agree on the policy contents. For example, in a personal auto policy, the applicant can select different levels of coverages to see the difference in premium. Select **New Version** in the submission wizard to create a new version of the submission that you can change and obtain a quote on. You can also make a new version based on other versions.

Under the **Tools** menu, click **Policy Versions** to display the **Policy Versions** screen where you can:

- Rename your version (for convenience).
- Click **Diff** to compare the differences between two versions.
- Make one version the selected version.
- Withdraw a version, while keeping the other versions.

Closing a Submission

You have the option to close a submission by selecting **Withdraw Transaction**, **Decline**, or **Not Taken** under **Close Options** in the submission wizard. Having separate closing options for a submission that was not bound allows you to track information such as how many were not bound or why they were not bound.

Withdraw Transaction – A submission can be withdrawn for any reason, such as mistakes were made on the policy. You can only withdraw a submission in Draft or Quoted status. The withdrawn submission, and all its versions are no longer editable.

Decline – An underwriter may decide to decline a submission, and if so, must provide a reason by entering a **Reason Code**. A submission may be declined for reasons including loss history, payment history, or requested coverages and/or limits not available. The declined submission is no longer editable.

Not Taken – Select this option when the applicant decides not to take the offered policy. You must enter a reason. You can also enter text to create a **Not Taken** letter. You can select this option from both the submission wizard and the **Submission Manager**. Generate the **Not Taken** letter from the **Submission Manager**.

Differences Between Binding a Policy and Issuing a Policy

PolicyCenter makes a distinction between binding and issuing a submission.

Binding means that the insured and the carrier have agreed to terms and price and a policy is in force. If the insured has a car accident one hour after the submission is bound, the policy covers the incident even though the insured has not received official documents.

Issuing means that a submission may be bound, but its policy documents have not yet been issued. Perhaps the carrier needed to collect or verify additional information so the issuance of the policy occurs after adding the information to the policy.

In PolicyCenter you:

- Bind a submission by clicking **Bind Only** under **Bind Options** in the user interface.
- Issue a submission in the submission wizard by clicking **Issue Policy** under **Bind Options** in the user interface.
- Issue a submission at a later date through an issuance policy transaction. See to “Issuance Policy Transaction” on page 89 for more information.

Expiring Submissions

A submission is expired after a sufficient and configurable interval of time has elapsed. A policy version can be expired when its status is either New, Draft, or Quoted. When a policy version expires, its status changes to Expired. You can view the status in the **Submission Manager** screen or in the toolbar if you are in the submission wizard. An expired branch is not editable.

See also

- “Changing Jobs to Expired Status” on page 551 in the *Configuration Guide*

Copying Submission Information

In PolicyCenter, you can copy submission information from an existing submission to create a new submission. Why? After losing the business to a competitor, a producer may copy submissions that were not taken and try to capture that business again. Another reason to copy submissions is if you need to reenter information for a new submission. Copying saves you data entry time and lessens the chance of errors. You can copy a submission with any status. If you copy a quoted submission, then PolicyCenter invalidates the quote (you must quote again) and the submission is in Draft status. If you copy a submission which is bound, then the newly copied submission is also in Draft status. However, you can copy only one version of a multi-version submission. This version is the version on view in the user interface when you select **Copy**.

If you select **Copy** from the **Actions** menu, the Submission Wizard creates a new submission which is based on the original, but with a new submission number. This new submission contains the same policy information, producer, underwriter company, and effective and expiration dates as the original. You can change any of that information. However, PolicyCenter does not copy the pre-qualification status, underwriter approval, and all financial information, so you must enter that information and obtain a quote prior to binding.

See also

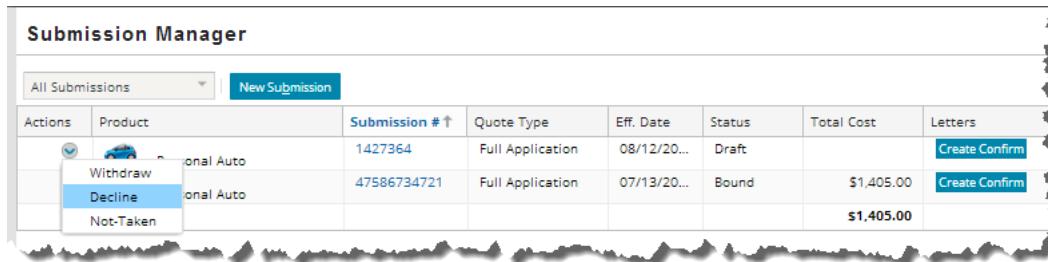
- See “Copying a Submission” on page 88 to see an example of how to copy a submission.

- See “Configuring the Copy Submission Feature” on page 561 in the *Configuration Guide*
- “Multi-version Quoting” on page 169

The Submission Manager

The **Submission Manager** screen contains summary information such as line of business, quote type, effective date, status of the transaction, and the premium. You can use this screen to do actions, such as withdrawing a submission. Use the **Submission Manager** is to view multiple submissions on an account and view the aggregate premium of all policies on the account. the **Submission Manager** has the following filters: **All Submissions**, **Open Submissions**, and **Complete Submissions**. **All Submissions** is the default. If you select **Open Submissions**, then the screen displays only submissions that have at least one version in an open status. If you select **Complete Submissions**, then the screen displays only submissions with no open versions. Therefore, a submission that has an open version appears under **Open Submissions** but not under **Complete Submissions**.

From the **Submission Manager** → **Actions** drop-down list, you can withdraw, decline, or not take a policy as shown in the following illustration. This drop-down list applies to both Quick Quote and Full Application. See “Closing a Submission” on page 84 for additional information on these actions.



Working with Submissions

This topic explains, from a user’s point of view (whether they are producers or underwriters), how to create the various types of submissions.

- “Places to Create a Submission” on page 86
- “Creating a Submission” on page 87
- “Copying a Submission” on page 88

See also

- “Policy Tools Menu” on page 307
- “Creating and Comparing Multi-version Quotes” on page 170

Places to Create a Submission

PolicyCenter has several places in the user interface where you can create a submission. However, you must have an account before you create a submission. Generally, if you have not selected an account and want to create a submission, PolicyCenter guides you to select an account first. If you already have an account, you then confirm the producer, product, quote type, and number of desired policies. You can create a submission from:

- Desktop → Actions
- Account File → Actions
- The Policy tab drop-down list
- The **Submission Manager** screen

- Copying another submission

Creating a Submission

The following steps explain the process for creating a submission whether it is a Quick Quote or a Full Application. Other steps are variations on this basic process.

Note: You must have an account before you can create a submission. To learn about accounts, see “Account File” on page 323.

1. If an account already exists, you can begin a submission by selecting the account from the **Account** tab or from the **Desktop** tab → **My Accounts** menu.
2. In the **Account File Summary** screen, click **Action** → **New Submission**. Since you have already defined an account, PolicyCenter directs you to the **New Submissions** screen which has default values for producer, producer code, and date.
3. Under the **Product Offers** section, select either **Single** or **Multiple**. Selecting **Multiple** allows you to enter the number of submissions you want to create. In the default application, you can create up to a maximum of five per line of business. You can configure this maximum in Studio. In this example, you create only one submission, so click **Select**.
4. Select the type of quote you want (**Quick Quote** or **Full Application**), then choose from the available product offerings. In this example, select **Full Application**.
 - **Quick Quote** gathers the minimal information needed to generate a quote.
 - **Full Application** gathers the complete set of information needed to apply for a bindable quote.
5. Depending on the line of business, there may be Pre-Qualification questions that need to be answered. If answered successfully, then the next screen is the **Policy Info** screen.
6. The **Policy Info** screen allows you to collect information, determine policy details, and if you have the correct permissions, change producer information and underwriting companies. You can also create or add other contacts as named insureds on the policy. These named insured contacts might be a spouse or child of the primary named insured. Named insureds can be a company, a person, or selected from an address book.
7. Continue entering required information in the submission wizard. Each line of business can have specific requirements that need to be captured in the policy. For example, a workers’ compensation policy can require additional information on locations, coverages, supplemental information, and workers’ compensation options.
 - See “Workers’ Compensation” on page 279 for more information on workers’ compensation.
 - See “Personal Auto” on page 257 for more information on personal auto.
 - See “Businessowners” on page 191 for more information on businessowners.
8. After entering the required information, you can review it in the **Policy Review** screen. If you are satisfied, then click **Quote**.
9. After the submission has been successfully quoted, you can:
 - Edit and requote. To edit submissions, you must have the **viewsubmission** and **editsubmission** system permissions.
 - Create a new version.
 - Save the draft.
 - Select from close options.
 - If you selected **Quick Quote**, then you can select **Full Application** to continue to enter additional information and bind the policy.
10. *(Optional)* Select **Forms** to view the list of forms that will be attached to the policy.

11. Select **Payment**. Enter the type of billing plan, payment type, and the deposit collected on the **Payment** screen.
12. Select a bind option from **Bind Options**:
 - **Bind Only**, legally binds both the carrier and the applicant, generates billing information, but does not issue the policy.
 - **Issue Policy**, binds, generates billing information, and issues the policy.

Copying a Submission

You can copy information from a submission to create a new submission. For additional information on this see “Copying Submission Information” on page 85.

1. Find the submission you want to copy and navigate to the **Submission Policy Info** screen.
2. From the **Actions** menu, select **Copy**.
Make any changes beginning with the **Policy Info** screen.
3. After you finish, you can create a new version, save the draft, select from the bind options, or select from the close options.

Issuance Policy Transaction

A submission may be bound, but its policy documents may not yet be issued because the carrier may need to collect or verify additional information. The mechanism that PolicyCenter uses to support this final step is the issuance policy transaction.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

This topic includes:

- “Issuance Overview” on page 89
- “Issuance General Steps” on page 90
- “Working with Issuances” on page 91

See also

- “Configuring Issuance” on page 563 in the *Configuration Guide*

Issuance Overview

There are times that you may choose to bind a submission without issuing it. Perhaps you need to collect additional information that binding does not require but issuing the final policy contract requires, such as:

- Verification of eligibility for discounts in an auto policy.
- Name and address of the additional interest because the insured does not own the vehicle (but the bank does).
- Receipt of VIN (Vehicle Identification Numbers) for vehicles on a business auto policy.

The point is, that while the carrier has agreed to provide coverage, there are some details that must be confirmed before generating the paperwork.

How Issuance Works

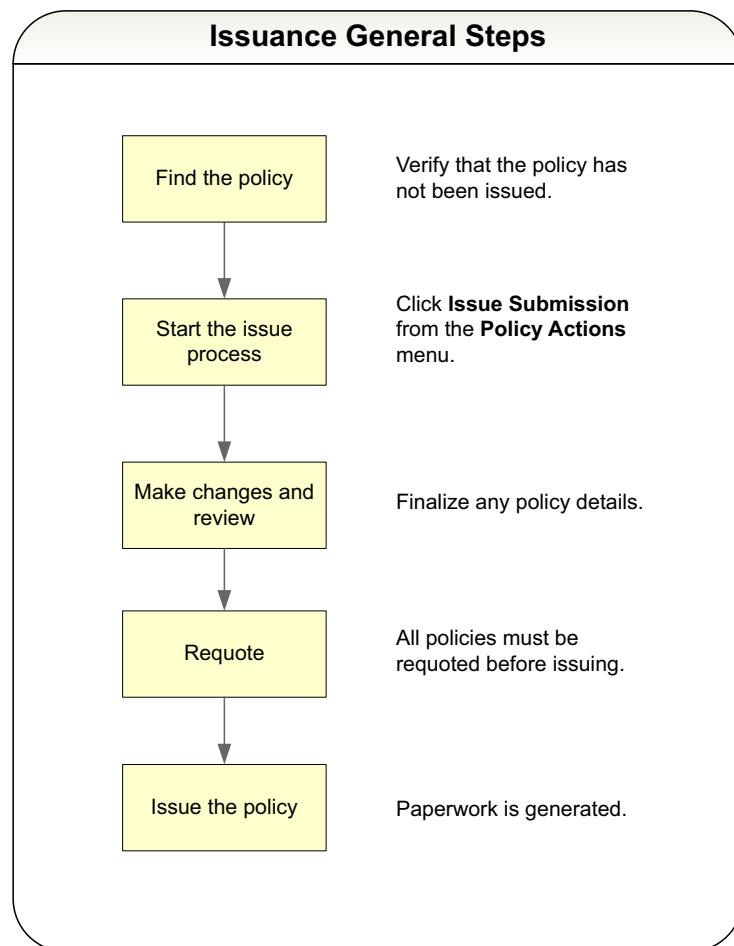
To start an issuance policy transaction, the policy must already be bound, but not issued. In addition, the policy must be no other open policy transactions. The issuance wizard behaves in much the same way as the submission wizard. You can edit in the **Policy Info** screen, review the policy, requote, and change payment options. To issue a policy, you must have the **bindissuance** system permission. A carrier can use this permission to specify the set of users who can issue policies. This set may differ from the set of users who can create policies.

Validation rules run at the Ready for Issue validation level. If validation fails with errors, then the process stops and remains in the previous Quoted status. If validation fails with warnings the first time, you can override the warnings by clicking **Issue Submission** again. Before binding the policy, PolicyCenter first executes evaluation and then sends the Issue Submission message.

After a policy has been successfully issued, PolicyCenter sends billing instructions to the billing system through the **IBillingSystemPlugin**.

Issuance General Steps

The following diagram indicates the basic steps to issue a policy. These steps are for the base configuration of PolicyCenter. Your business requirements and even lines of business can alter the process.



Working with Issuances

This section describes how to work with issuance policy transactions in the user interface.

See also

- “Policy Tools Menu” on page 307

Issuing a Policy

To issue a policy, you must first have a submission that has been bound but not issued.

1. Navigate to the **Policy File Summary** screen.
2. From the **Policy** tab **Actions** menu, select **Issue Submission**.
3. Beginning with the **Policy Info** screen, make any necessary changes. You can make any changes to the policy, including changing the effective and expiration dates.
4. You must requote the policy. Click **Quote**.
5. Click **Issue Submission**. A dialog box verifies your action. PolicyCenter sends the policy information to be generated and mailed by a print issuance system. The outcomes are:

If **PolicyCenter successfully issues the policy**, then the **Issuance Bound** confirmation screen appears and the status is set to **Bound**. Your options are to:

- View your issuance policy transaction.
- View your policy.
- Go to the **Submission Manager** for the selected account.
- Submit an application for a different account.
- Go to your **Desktop**.

If **issuance fails**, PolicyCenter sets the **UWApproval** to **Review** status and creates an **Issuance failed** activity. It then assigns it to the underwriter.

Renewal Policy Transaction

Carriers typically begin the process of renewing a policy for another period of time before its expiration. The most efficient way is to have PolicyCenter process these renewals automatically. However, sometimes an underwriter or producer must review or make adjustments to the policy before deciding whether it can be renewed. PolicyCenter is flexible in handling both automatic and manual renewals.

This topic describes the renewal policy transaction in PolicyCenter, how to work with it.

This topic includes:

- “Renewal Overview” on page 93
- “Renewal General Steps” on page 96
- “Renewal Key Features” on page 98
- “Working with Pre-renewal Directions” on page 101
- “Working with Underwriting Issues and Referral Reasons” on page 103
- “Working with Renewals” on page 103

See also

- “Configuring Renewals” on page 565 in the *Configuration Guide*

Renewal Overview

Typically, as a policy nears its expiration date, carriers wishing to keep that business offer the policy again for another policy period. A renewal policy transaction extends a policy for another period of time.

The goals of renewal processing are to:

- Maximize retention of the best customers of a carrier.
- Reduce expenses associated with the renewal process.

For both the producer and the carrier, renewing an existing customer is more profitable than acquiring a new customer with a similar profile because of acquisition and processing costs. A carrier’s retention ratio (the percentage of insurance policies that renew) is a closely watched metric. Too low a retention rate might indicate

poor customer service to producers, noncompetitive pricing, or unfavorable claim service. Because the bulk of business is from existing customers, having an efficient renewal process has a great impact to minimizing overhead and optimizing revenue.

By design, PolicyCenter handles renewals efficiently. In the default configuration, policies nearing the end of their term are examined. If a policy can be renewed without requiring an underwriting decision, the renewal progresses automatically. However, if manual intervention (review of the policy) is necessary, then PolicyCenter guides you through this process.

An underwriter may need to review a policy for a variety of reasons, including:

- Manual rating required
- Carrier practices for that class of customer
- Unfavorable claims or payment history
- Significant changes in risks or exposures

PolicyCenter can track these variables through business rules evaluation, referral reasons, or pre-renewal directions which stop a policy from automatically renewing. This tracking allows an underwriter to review and make a decision on whether to renew, modify, or decline the policy.

This topic includes:

- “Starting Renewals Manually in the User Interface” on page 94
- “Starting Renewals by Using the Policy Renewal Web Service” on page 94
- “Starting Renewals by Using a Batch Process” on page 94
- “Renewal Flows” on page 95
- “Renewals Create New Policy Periods” on page 95
- “Renewal Restrictions” on page 96
- “Renewal Outcomes” on page 96

Starting Renewals Manually in the User Interface

There may be times that you need to start the process manually. For example, you may want to start a renewal policy transaction earlier than the predetermined number of days. Or you may want to start a renewal if the insured originally declined the renewal, then changed their mind, and now requests that their policy be renewed.

To start a renewal through the user interface, find an in-force policy, then select **Renew Policy** from the **Actions** menu. The renewal wizard guides you through the necessary steps.

See also

- “Creating a Manual Renewal” on page 104

Starting Renewals by Using the Policy Renewal Web Service

Renewals can be started by an external system through the policy renewal web service. The web service provides methods to start renewals on existing policies and to import a policy to PolicyCenter and start a renewal. The web service also provides methods for renewals in a billing system.

See also

- “Renewal Web Services and Plugins” on page 568 in the *Configuration Guide*

Starting Renewals by Using a Batch Process

PolicyCenter has a batch process that automatically finds policies that are ready for renewal. In the default configuration, PolicyCenter starts renewals based upon the expiration date, the renewal process lead time, line of

business, jurisdiction, and time of year. PolicyCenter first checks whether the expiration date of the policy period falls within the renewal process lead time. Then PolicyCenter determines the lead time required by regulations. PolicyCenter adds additional time for company practices. Finally, PolicyCenter adds a delay for concurrent policy transactions, if any. Because the renewal process lead time is checked first, no policy will start automatic renewal sooner than this.

Factors to Consider in Scheduling the Renewal Batch Process

There are several factors to consider in the schedule of the renewal batch process. Some of these factors are:

- **Frequency** – Suppose you have a batch process that runs every Sunday. A renewal that could start on Wednesday will not start until the batch process runs several days later on Sunday.
- **Start time** – The batch process starts at 11 p.m. on Sunday, and it may take several hours to run. Therefore, the batch process starts some renewals on Sunday and others early Monday morning.
- **Postal service** – The renewal paperwork depends upon the working days of the postal service. Although the renewal was started on Sunday, the postal service picks up the renewal paperwork on Monday.

See also

- “Configuring Batch Process Renewals” on page 570 in the *Configuration Guide*

Renewal Flows

PolicyCenter supports the following renewal process flows:

- Bind and cancel
- Renewal offer
- Confirmed renewal

In the default configuration, PolicyCenter uses the *bind and cancel* renewal flow for all lines of business. When you bind a renewal, PolicyCenter sends charges to the billing system. PolicyCenter then does a flat cancel for reason *Policy not taken* if no payment is received for that period. If partially paid, then PolicyCenter cancels for reason *Non payment*.

The default configuration contains the *renewal offer* renewal flow which binds only after payment. You can configure this renewal flow for a particular line of business. Under this approach, you make the decision to renew or not renew, but instead of actually binding the renewal, you consider it a *renewal offer*. When you make the renewal offer, PolicyCenter sends a renewal notice (including pricing and payment plans). PolicyCenter does not send charges to the billing system (since no policy transaction has been completed). When the billing system receives payment, it sends a message to PolicyCenter to bind the renewal. If payment is not received, the PolicyCenter renewal flow times out. The renewal is considered not taken.

The default configuration contains the *confirmed renewal* flow which provides confirmation from the billing system that the insured has completed payment. PolicyCenter knows if the policy was confirmed and is legally binding. The *bind and cancel* flow does not provide either of these.

See also

- “Billing Integration” on page 509 in the *Integration Guide*

Renewals Create New Policy Periods

The renewal policy transaction creates a new policy period with new effective and expiration dates.

The only restriction is that the effective date of the renewal policy must not overlap with the policy period of the current in-force policy.

Renewal Restrictions

Any policy that has been issued and is in-force can be renewed, however there are limitations to starting a renewal.

- There can be no open rewrite policy transactions on the policy.
- There can be no open renewals on the policy.
- The policy cannot be canceled.

Renewal Outcomes

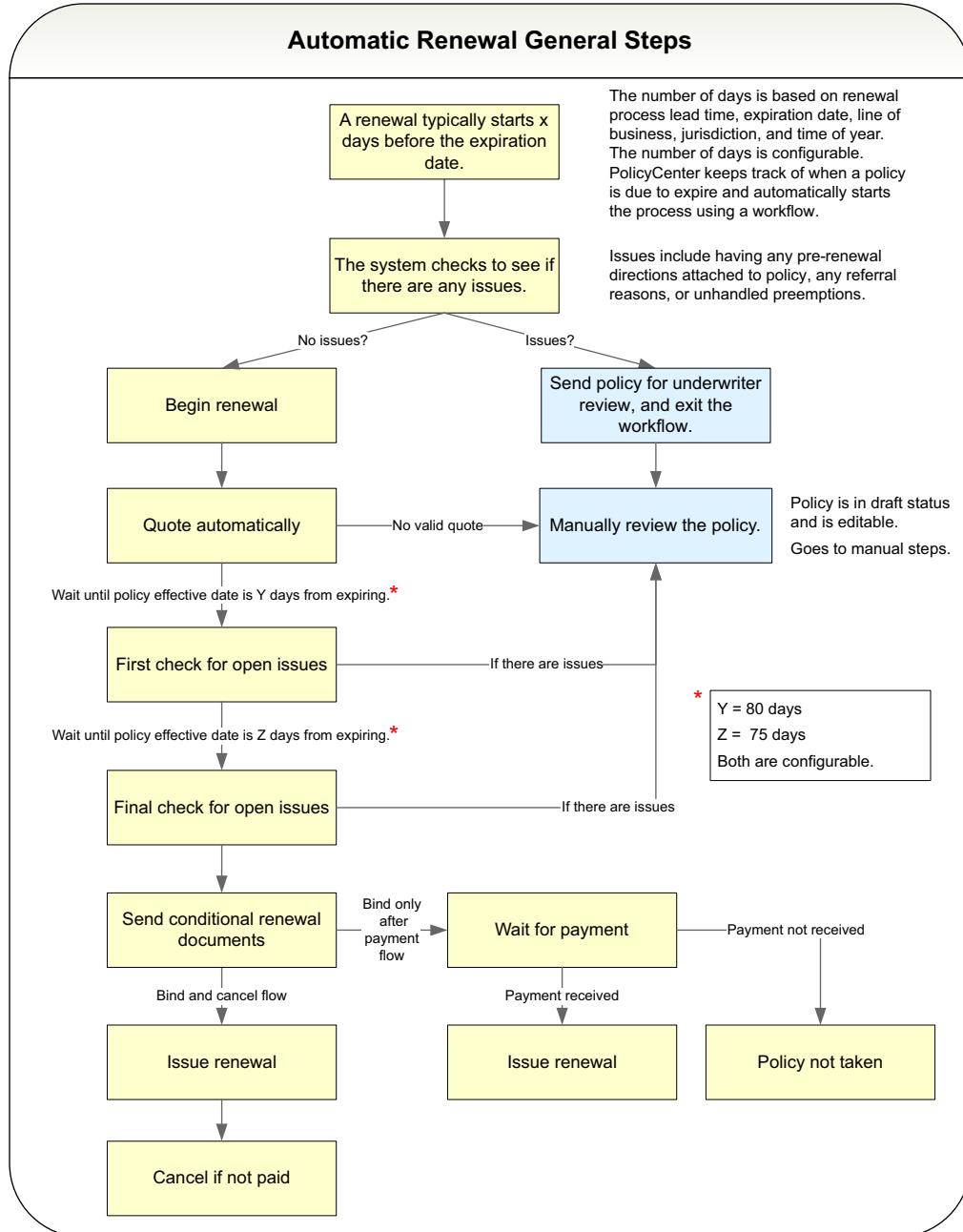
Renewals can have one of the following outcomes:

- **Renewed** – A policy is renewed for another period of time.
- **Not Taken** – The insured declines the offered policy, and PolicyCenter marks the renewal as not taken.
- **Not Renewed** – The carrier decides not to renew the policy, and the policy expires on the expiration date.

Renewal General Steps

The diagrams in this topic show the basic steps used in an automatic and manual renewal policy transaction in the default application. You can customize the flows based on your business requirements.

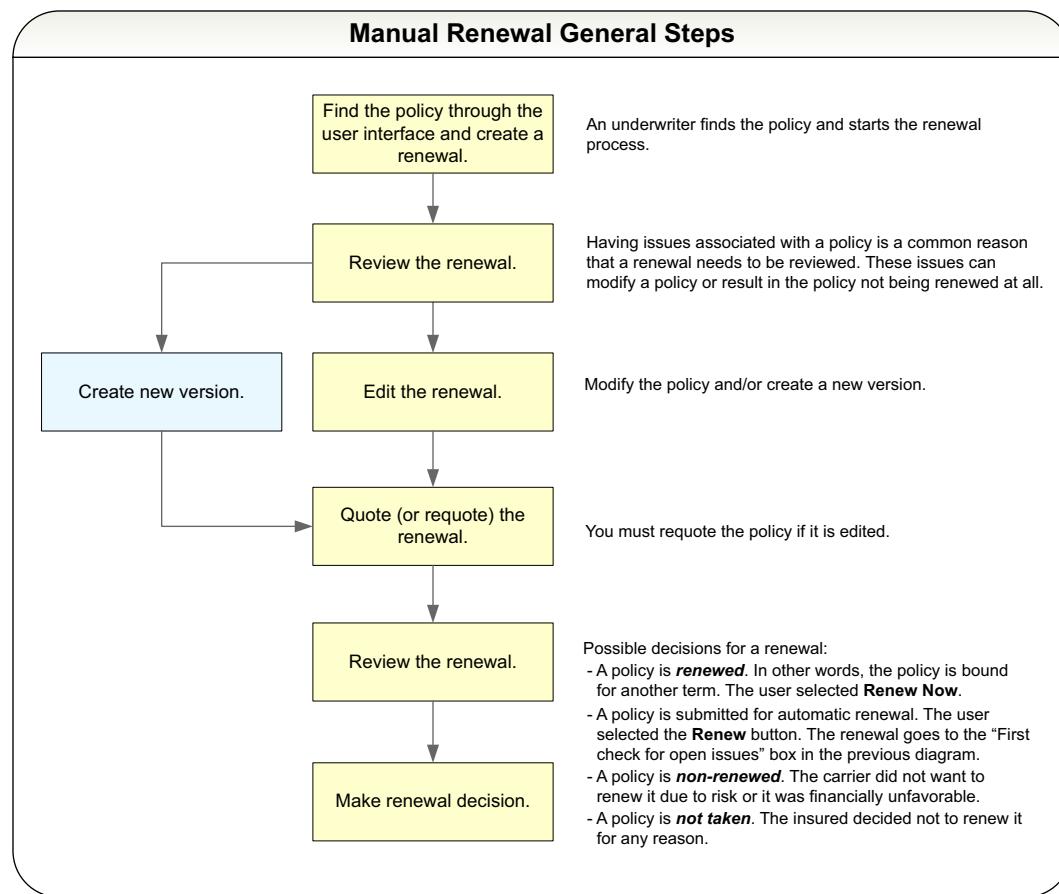
PolicyCenter uses an automatic process to renew a policy without human intervention. To view the default steps, see “Renewal Workflows” on page 567 in the *Configuration Guide*.



The two checks for open issues occur 80 and 75 days before the policy expires. You can configure these in the `PendingRenewalFirstCheckDate` and `PendingRenewalFinalCheckDate` methods in `RenewalProcess.gs`.

An underwriter may use the manual process if the renewal needs modification and the renewal needs to be started before the scheduled time for the renewal batch process. An underwriter may also use the manual process if a renewal was previously declined (not taken), then the insured changed their mind, and now wants the policy renewed.

The following example does not factor any issues that may need to be dealt with first.



Renewal Key Features

Key features of renewal policy transactions in PolicyCenter are:

- “Understanding Pre-renewal Directions” on page 98
- “Understanding Referral Reasons” on page 99
- “Understanding Underwriting Issues” on page 99
- “Understanding Preemption” on page 100
- “Creating Multiple Versions” on page 100

Understanding Pre-renewal Directions

A pre-renewal direction is a special type of note which indicates how to handle the renewal. PolicyCenter attaches this special note to a policy, but you cannot view the note in the user interface. Creating these directions can save the underwriter from revisiting the renewal policy transaction at a later time.

Note: If you want the policy to be automatically processed, do not create a pre-renewal direction that assigns renewals to a user.

Pre-renewals have the following broad directions:

- **Non-renew** – Indicates not to renew the policy.
- **Not taken** – Indicates that the insured did not take the renewal policy.

- **Refer to an individual for review** – Indicates that a person needs to manually review the policy before deciding its outcome. This person can be an underwriter, a customer service representative, or an underwriter assistant.

Usually, a user who knows how to handle the renewal creates the pre-renewal direction.

Examples

- The policy has become high risk, so an underwriter now must review it.
- The claims department finds that the policy has too many outstanding claims, therefore the insurer will not renew the policy.
- The insured contacts the producer and indicates that a better rate can be found through a competitor, so the insured will not take the policy.

A policy can have, at any given time, only one active pre-renewal direction. If a policy has a pre-renewal direction, then the renewal process uses this pre-renewal direction.

Note: You cannot create pre-renewal directions for a policy period which has already been renewed. Pre-renewal direction cannot be set/edited on a policy if there is a renewal on the active policy.

See also

- “Working with Pre-renewal Directions” on page 101

Understanding Referral Reasons

You can use referral reasons to record underwriting issues for a policy outside of the context of a policy transaction. Because you add referral reasons to the policy, you can create referral reasons at any time, even when there is no policy transaction. PolicyCenter creates underwriting issues for these referral reasons the next time a user processes a policy transaction on the policy. It also checks for referral reasons at various point in the policy transaction progress. Referral reasons can block progress of policy transactions on the policy. In the default application, the renewal policy transaction checks to see if there are any open referral reasons in the prior term.

Referral reasons can be added manually through the user interface or programmatically by using the `addReferralReason` method in `PolicyEnhancement.gsx`.

Note: Referral reasons affect all policy transactions that handle underwriter issues, not just renewals.

See also

To obtain detailed information on referral reasons:

- “Underwriting Referral Reasons on the Policy” on page 412
- “Configuring Underwriting Referral Reasons” on page 461 in the *Configuration Guide*.

To learn how to create a referral reason and view them:

- “Working with Underwriting Referral Reasons” on page 416

Understanding Underwriting Issues

Underwriting issues allow you to track issues with a policy version that are of interest to an underwriter. Among other things, each issue can specify:

- An issue type
- A description
- A point in the policy transaction where the issue is raised
- A point where the issue blocks progress

Underwriting issues are raised based on the underwriting authority of the current user. Issues are sent for approval to a user with greater underwriting authority. Open issues must be reviewed and approved before a renewal can be successfully completed. You can create underwriting issues in the following ways:

- **Add an underwriting issue through the user interface.** You can only do this in the context of policy transactions. In the policy transaction, click Risk Analysis from the Tools menu, and select the Underwriting Issues tab. You can add a new issue, close an existing issue, or reopen a closed issue. When creating a new issue, you select a type and, optionally, describe the issue.
- **Create issues automatically in the Evaluation rule sets** in Guidewire Studio. When PolicyCenter calls the Evaluation rule sets at various points during renewal, the system identifies and generates underwriting issues.
- **Create a referral reason on the policy.** PolicyCenter creates underwriting issues from referral reasons at the start of a renewal policy transaction and at various points during the policy transaction. Use referral reasons for issues that apply to the policy as a whole, not just the policy period.

Automated renewal policy transactions usually approve issues automatically based on the underwriting authority of an automated renewal user that you specify.

See also

- “Understanding Referral Reasons” on page 99
- “Underwriting Authority” on page 411
- “Automated Renewals” on page 413
- “Configuring Underwriting Issues” on page 457 in the *Configuration Guide*

Understanding Preemption

A renewal is preempted when some other policy transaction (for the in-force period) is bound and affects the renewal period. Preemptions are not unique to renewals; they also occur in audits, cancellations, reinstatements, and policy changes. PolicyCenter handles these types of changes by letting you know when certain policy transactions are not sequential. Through the user interface, PolicyCenter provides you with a way to see the differences so that you can resolve them. You can configure how PolicyCenter handles preemptions.

See also

- “Preempted Jobs” on page 503

Creating Multiple Versions

You can create multiple versions of a renewal to help you and the insured agree on the policy contents. For example, in a personal auto policy, the applicant may select different levels of coverages to see the difference in premium. Selecting **New Version** in the renewal wizard creates a new version of the renewal where you can make changes and obtain a different quote.

Under the Tools menu, click **Policy Versions** to display the **Policy Versions** screen where you can:

- Rename your version (for convenience).
- Click **Diff** to compare the differences between two versions.
- Make one version the selected version.
- Withdraw a version, while keeping the other versions.

See also

- “Creating a Manual Renewal” on page 104 for an example.

Working with Pre-renewal Directions

This topic describes how to work with pre-renewal directions and contains the following:

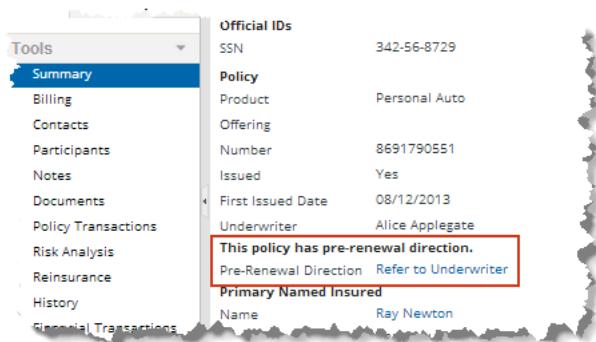
- “Viewing an Existing Pre-renewal Direction” on page 101
- “Creating a Pre-renewal Direction” on page 101
- “Deleting a Pre-renewal Direction” on page 102

Note: There can only be one pre-renewal direction on a policy.

Viewing an Existing Pre-renewal Direction

You can view a pre-renewal direction on a policy.

1. In this example, find a policy that you know has a pre-renewal direction.
2. Click the **Summary** link under the Tools menu. A policy that does not have a pre-renewal direction, does not display this information because the pre-renewal direction widgets are not visible.



Creating a Pre-renewal Direction

1. Navigate to a policy.
 2. Select **Pre-Renewal Direction** from the **Actions** menu to view the **Pre-Renewal Direction** screen. If the policy has no pre-renewal direction, then the **Details** in the pre-renewal direction screen is initially blank.
 3. To create a pre-renewal direction or modify an existing one, click **Edit**.
 4. Select a direction from the drop-down list, add text, and set the security level which controls who can view it.
- In the default configuration, you can specify that the renewal:
- Ends in non-renewal
 - Ends in not taken

- Be referred to a customer service representative, underwriter, or underwriter assistant

The screenshot shows the 'Pre-Renewal Direction for Policy Term' screen. On the left, there's a sidebar with 'Actions' and 'Policy Contract' sections. The main area has tabs for 'Update', 'Cancel', and 'View Notes'. Under 'Details', the 'Direction' dropdown is set to '<none selected>' and is highlighted with a red box. Below it, a list of options includes 'Non-Renew' (which is also highlighted with a red box). Other options like 'Non-Renew and Refer to Underwriter' and 'Refer to Customer Service Representative' are listed. To the right, there's a 'Note' section with a 'Topic' dropdown set to 'Pre-renewal direction' and a 'Security Level' dropdown set to 'Internal Only', both of which are also highlighted with red boxes.

Note: Creating or changing a pre-renewal direction adds an entry to the history of the policy.

5. If you selected a non-renew direction, you can add a non-renewal explanation.

The screenshot shows the 'Pre-Renewal Direction for Policy Term' screen again. The 'Direction' dropdown is now set to 'Non-Renew' (highlighted with a red box). Below it, the 'Non-Renew Reason' dropdown is set to '<none selected>'. In the bottom right corner, there's a separate panel titled 'Non-Renewal Explanations' with a link 'Return to Pre-Renewal Direction'. This panel has 'Add' and 'Cancel' buttons. It lists several explanations, with the first one checked: 'This policy type, Work...'. There are other unchecked options like '1604782547 loss was ...' and 'The primary insured, ...'.

See also

- “Configuring Explanations in Pre-renewal Directions” on page 572 in the *Configuration Guide*

Deleting a Pre-renewal Direction

Pre-renewals can be removed.

1. Navigate to a policy.
2. Navigate to the **Summary** screen as in the previous example.
3. Click the **Pre-Renewal** link. The **Pre-renewal Direction for the Policy Term** screen appears.

4. Click **Delete**.

Working with Underwriting Issues and Referral Reasons

Instructions on how to work with underwriting referral reasons and underwriting issues are provided in:

- “Working with Underwriting Issues” on page 414
- “Working with Underwriting Referral Reasons” on page 416

Working with Renewals

This topic explains how to work with renewals from a user’s point of view.

- “Creating a Renewal from a Batch Process” on page 103
- “Creating a Manual Renewal” on page 104
- “Viewing Your Renewals” on page 104

See also

- “Policy Tools Menu” on page 307
- “Working with Multi-version Quoting” on page 169

Creating a Renewal from a Batch Process

Creating a renewal from a batch process is the most common way for renewals to be created. In a production environment, the renewal batch process looks for policies that expire in X days (the default is 165 days). Then the Policy Renewal plugin evaluates each policy and determines whether to creates a renewal for that policy. A system administrator can also run this batch process outside of its schedule.

The frequency of the batch process is a factor in when a renewal starts. For example, you can have a batch process that runs every Sunday. Renewals that could start on Wednesday will not start until the batch process runs on Sunday, several days later.

Note: You must have the correct permissions for this process. Typically, you must have system administrator or super user permissions.

1. In PolicyCenter, type Shift + Alt + T to display **Server Tools**.

2. Select the **Batch Process Info** link in the left sidebar.

The **Batch Process Info** screen contains useful information about the batch process, including:

- Current status
- The last time it ran
- The time of the next scheduled run
- The schedule

The **Cron-S M H DOM M DOW** column header stands for seconds, minutes, hours, days of month, month, and day of week.

Note: The * means *every*. The ? is typically only on day of week or day of month and means, “I do not care when it runs”.

3. Find **Policy Renewal Start** under the **Batch Process** column.

4. Click **Run** under the **Action** column to start the batch process immediately.

See also

- The `scheduler-config.xml` file to see the frequency of the batch process. You can view this file by navigating to **configuration** → **config** → **scheduler** in Studio.
- The `config.xml` file for the lead time parameter. This parameter contains the number of days before the policy expires and renewal processing starts. You can view this file by navigating to **configuration** → **config** in Studio.

```
<param name="RenewalProcessLeadTime" value="165" />
```
- “Configuring Batch Process Renewals” on page 570 in the *Configuration Guide*

Creating a Manual Renewal

This example is useful if you need to create a renewal before its scheduled time and need to make changes to the policy.

1. Find the policy you want to renew.
2. From the **Policy Info** screen, select **Actions** → **Renew Policy**. A dialog box asks you to confirm your selection and the renewal wizard begins.
3. Click **Edit Policy Transaction**. Make the required changes to the policy.

At this point, you can click one of the following:

- **Quote** – You must generate a quote. If the policy had not been edited, the renewal wizard would perform the quote step and obtain a new premium.
- **Save Draft** – You can return to work on it at a later time.
- **Close Options** → **Withdraw Transaction** – You can withdraw the renewal.
- **Close Options** → **NonRenew** – The policy is not renewed, and PolicyCenter asks you to give a reason.
- **Close Options** → **Not Taken** – Use if the insured declines to renew the policy.

4. In this example, click **Quote**.
5. On the **View Quote** screen, you have various options, but first review the quote and the policy in general.

6. In this example, you use the original modified policy. Your options are:

- **Edit Policy Transaction** – Put the policy in **Draft** status. You can edit and requote again.
- **Bind Options** → **Renew** – Puts the policy transaction into a **Renewing** state, so that it will eventually be completed with an outcome of **renewed**, but not until the actual effective date of the renewal
- **Bind Options** → **Issue Now** – Immediately finish the policy transaction and renew the policy.
- **Close Options** → **Withdraw Transaction** – Remove the renewal from consideration.
- **Close Options** → **NonRenew** – Complete the renewal process by setting the outcome to **NonRenewed** and the policy expires. You must give a reason such as insured request, losses, or payment history.
- **Close Options** → **Not Taken** – Complete the renewal process by setting the outcome to **NotTaken**. The policy expires.
- **Versions** → **Start Multi-Version** – Create new versions of the policy. For more information, see “Creating and Comparing Multi-version Quotes” on page 170.
- **Print Quote** – Print the quote.

Viewing Your Renewals

The **My Renewals** screen displays a renewal if:

- You have been assigned a role on that renewal.
- You have or had an activity assigned to you on that renewal.

To view your renewals

1. Go to the Desktop and select **My Renewals**.
2. You can select a group from the **Group** column or a policy from the **Policy Type** column. You can filter your search by selecting from the drop-down menu displaying **All open**.

Cancellation Policy Transaction

A cancellation policy transaction is the process of voiding a policy while it is in force. Initiated either by the carrier or the policyholder, it results in the policy:

- Being canceled
- Remaining in force because the cancellation was rescinded

An example of a cancellation policy transaction is when the policyholder does not pay the premium, so the carrier begins the cancellation process. The policyholder receives notice of a pending cancellation in the mail, contacts the producer, and explains that there was a billing mix-up and sends another payment to the producer. The policy was in the process of being canceled but had not completed cancellation. Upon receipt of payment, the carrier rescinds the cancellation, and PolicyCenter withdraws the cancellation.

Another example of cancellation is when the carrier cancels the policy. The carrier issues a liability policy to a restaurant for two locations with 10% liquor sales. An audit reveals that the restaurant actually has four locations with 80% liquor sales. The carrier begins the cancellation process and, after a set number of days, the policy is canceled.

The insured can initiate a cancellation. For example, the insured calls to cancel their businessowners policy because they are no longer in business.

A cancellation can be generated and rescinded automatically. For example, a billing system can initiate a cancellation for non-payment of premiums or rescind a cancellation after receiving payment.

A canceled policy can be reinstated. For more information, see “Reinstatement Policy Transactions” on page 125.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

This topic includes:

- “Cancellation Overview” on page 108
- “Cancellation General Steps” on page 111

- “Working with Cancellations” on page 111

See also

- “Configuring Cancellations” on page 575 in the *Configuration Guide*

Cancellation Overview

A policy cancellation ends the policy contract. A flat cancellation cancels the policy as of the policy effective date and voids the contract. Other cancellations are effective after the policy effective date but prior to the policy expiration date. The contract ends midterm.

A cancellation policy transaction can be started either manually through the policy file or programmatically. The selected source and reason determine if premium will be calculated pro rata or with penalties. The selected source and reason also determine the date on which the policy cancellation completes.

The source of a cancellation can be either the insured or the carrier. You can configure the cancellation reasons based on your business needs. In the default configuration, some of the reasons for cancellation by insured are:

- Policy not taken
- Out-of-business

In the default configuration, some of the reasons for cancellation by the carrier are:

- Fraud
- Failure to comply with terms and conditions
- Underwriting reasons
- Policy to be rewritten or replaced by company
- Or more commonly, non-payment

The source, reason, and effective date affect the premium calculation method. Premium calculation methods are:

- *Pro rata* – The carrier bills the policyholder for the time that the policy was already in effect.
- *Short rate* – The carrier charges the policyholder a penalty in addition to the pro rata amount.
- *Flat* – The carrier refunds the total amount of the policy.

Scheduling Versus Cancel Now

In the user interface, you have the option to select **Bind Options** → **Schedule Cancellation** or **Cancel Now**.

If you choose **Schedule Cancellation**, the cancellation is completed as of the cancellation effective date. The cancellation source, reason, and policy type determine the default cancellation effective date.

If you choose **Cancel Now**, PolicyCenter sets the cancellation completion date to the current date and issues the cancellation. Once the cancellation is issued, you cannot rescind it. If you want to undo the cancellation, you have to reinstate the policy. Although the cancellation is completed when the user selects **Cancel Now**, the cancellation does not go into effect until the cancellation effective date.

Until a cancellation is completed, it is considered an open cancellation.

See also

- “Calculating the Cancellation Effective Date” on page 578 in the *Configuration Guide*

Default Cancellation Effective Date

The carrier usually bases the default cancellation effective date on the type of cancellation, the regulations of the governing jurisdiction, and the line of coverage that is being canceled. For example, two policies in the same

jurisdiction may require different notification periods due to differing lines of business. A commercial property policy may have different notification periods in adjacent jurisdictions. Non-payment may require only 10 days notification in one jurisdiction, but 30 days in another. In very limited conditions, the carrier can also cancel immediately.

The system allows the user to override the default cancellation effective date in all circumstances, except in the case of a flat cancellation. For example, in a cancellation initiated by the carrier, suppose that at least 30 days must elapse before the cancellation becomes effective. The default cancellation effective date reflects the regulatory requirements for notifying a customer that their policy is being canceled. Therefore, the user can override the cancellation effective date only by moving it further out, allowing more days to elapse before the cancellation becomes effective. In most cancellations initiated by the insured, the cancellation effective date can be moved further in or further out. (The user must have the **Cancellation override effective date** permission.) The user can never override the cancellation effective date for a flat cancellation because the whole policy term has been canceled.

The following table shows how some choices are configured in the default installation.

Source	Reason	Refund Method	Default Cancellation Effective Date
Insured	Insured's request- (finance co. nonpay)	Pro rata	System's current date*
Insured	Insured's request- (N.O.C.)	Short rate	System's current date*
Insured	No employees/operations	Pro rata	System's current date*
Insured	Out of business/sold	Pro rata	System's current date*
Insured	Policy not taken	Flat	The date the policy went into effect.
Carrier	Cancellation of underlying insurance	Pro rata	Calculated based on jurisdiction and line of business.*
Carrier	Condemned/unsafe	Pro rata	Calculated based on jurisdiction and line of business.*
Carrier	Non payment	Pro rata	Calculated based on jurisdiction and line of business.*
Carrier	Policy rewritten (mid-term)	Pro rata	System's current date*
Carrier	Policy rewritten or replaced (flat cancel)	Flat	The date the policy went into effect.
Carrier	Fraud	Pro rata	Calculated based on jurisdiction and line of business.*

Note: “*” indicates that the user can override the default cancellation effective date.

See also

- “Calculating the Cancellation Effective Date” on page 578 in the *Configuration Guide*

Changing a Cancellation

There are a limited number of ways that you can change an existing cancellation. You can change the cancellation effective date of a policy. You can change the reason description.

You can make other types of changes to an open cancellation such as changing the source or reason. You can make these changes by withdrawing or rescinding the cancellation, or by scheduling an additional cancellation on the policy. If the cancellation notice has not yet been sent, you can withdraw the cancellation. If the cancellation notice has already been sent, you can rescind the cancellation. If the cancellation has already completed, you can reinstate the policy as described in “Reinstatement Policy Transactions” on page 125.

Canceling a Policy at an Earlier Effective Date

Because of regulatory requirements, you usually cannot change the cancellation effective date of an already scheduled cancellation to an earlier effective date. However, you can create a new cancellation on the same

policy with an earlier effective date if the regulatory requirements associated with the reason for that cancellation allow it. For example, you might have a policy that has an open cancellation for non-payment. The insured calls and asks to be canceled immediately. You create another cancellation on the policy, select **Insured** as the **Source** and select a **Reason**. On the **Confirmation** screen, select **Cancel Now**. This cancellation preempts all open cancellations on the policy. (When you bind a cancellation, PolicyCenter withdraws existing cancellations.)

If a policy has a completed cancellation, you can also cancel that policy on an earlier effective date. There can be multiple open scheduled cancellations (for different reasons) on the same policy.

Cancelling a Policy on the Same or Later Effective Date by Changing the Cancellation

There are several reasons for changing the cancellation effective date. For example:

- A policy holder receives cancellation notices stating that their policy will be canceled unless they submit payment. The policy holder contacts the carrier and asks for a few extra days to reestablish the policy by submitting payment. The agent reschedules the cancellation by adding three days to the cancellation effective date.
- A catastrophe takes place in a certain region. The carrier decides to give an extension on scheduled cancellations for all policies in or near the catastrophe. The carrier gives the extension to policies with a cancellation effective within a certain date range.

When you change the cancellation effective date, PolicyCenter sends a replacement cancellation notice. (You must configure this in PolicyCenter.)

If you have the **Cancellation reschedule** permission, you can change the cancellation effective date on an open cancellation. In the base configuration, underwriters, underwriter supervisors, and the superuser have this permission. An open cancellation is a cancellation that has not completed.

You can change the following fields on an open cancellation:

- **Reason Description** – You can provide a new description.
- **Cancellation Effective Date** – You can move the effective date.

You cannot change the following fields:

- **Source**
- **Reason**
- **Refund Method**

The cancellation effective date can be moved to a date on or after the earliest allowable cancellation effective date. The earliest allowable cancellation effective date continues to be based on the date the cancellation was originally scheduled. It is not based on the date that you make the change. Because this is just a change to the policy transaction, the original restrictions on the cancellation effective date remain the same. For more information, see “Calculating the Cancellation Effective Date” on page 578 in the *Configuration Guide*.

Other Ways to Cancel a Policy on the Same or Later Effective Date

It is possible to change the cancellation effective date to the same or a later effective date even if you do not have the **Cancellation reschedule** permission. If you have an open cancellation on a policy, create a new cancellation and issue it by selecting **Cancel Now**. (When you issue a cancellation, PolicyCenter withdraws existing cancellations.)

If the cancellation has already completed, but you want to cancel the policy at a later effective date, you can reinstate the policy, then cancel it. To learn about reinstatement, see “Reinstatement Policy Transactions” on page 125.

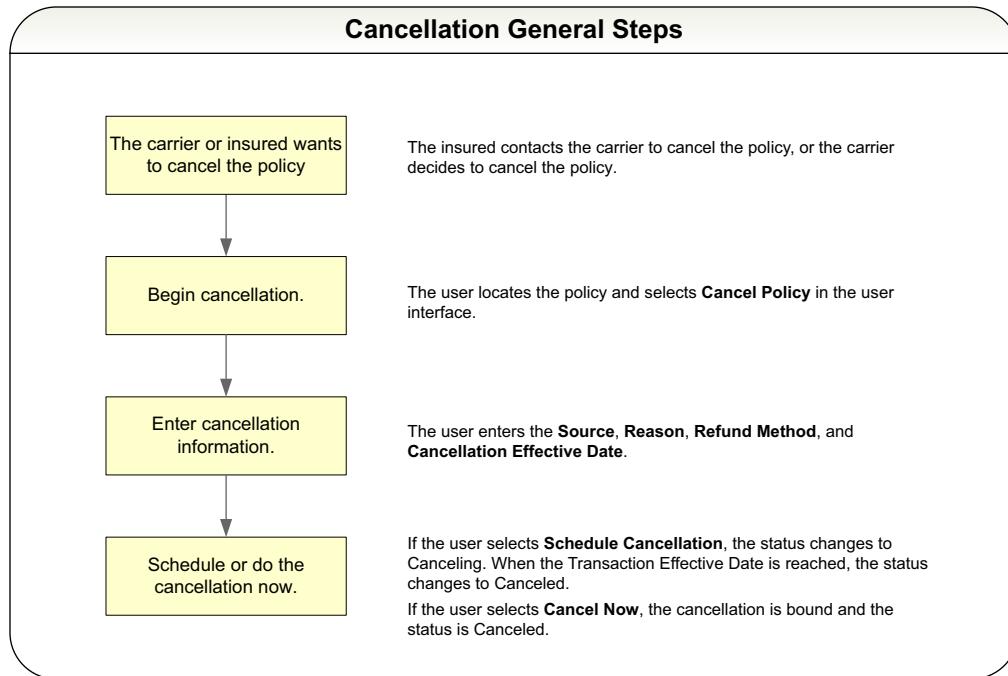
Rescinded Versus Withdrawn Cancellations

Rescinded and withdrawn cancellations have identical results: the outcome is that the policy remains in force. However, rescinded and withdrawn cancellations are two different actions. A cancellation may be withdrawn at

any time before cancellation notices are sent. If notices have already been sent, then the cancellation must be rescinded so that rescindment notices can be generated. One exception is if multiple cancellations were started for the same policy; if one of them completes successfully, then all the others are withdrawn. This exception occurs because the policy did get canceled, so no rescind notices need to be sent. Having different statuses for rescind and withdrawal can also be useful for statistical analysis and reporting.

Cancellation General Steps

The following diagram shows the basic steps in a manual cancellation in the default configuration of PolicyCenter. Your business requirements and lines of business can alter the process. For example, the steps may differ slightly between personal and commercial lines.



Working with Cancellations

This topic explains, from a user's point of view, how to work with cancellations in the user interface.

- “Canceling a Policy” on page 112
- “Rescinding a Cancellation” on page 112
- “Viewing Cancellations in Policy Transactions” on page 113

IMPORTANT The following describes a cancellation policy transaction in the default application, therefore, the following screen shots serve only as an example. Your configuration and line of business affect the look and flow.

See also

- “Policy Tools Menu” on page 307

Canceling a Policy

1. Navigate to the **Search** tab, find the policy to cancel, and select **Cancel Policy** from the **Actions** menu.
2. Enter the following information:
 - a. Select **Carrier or Insured** for **Source**.
 - b. Select a **Reason** from the drop down list.

The list varies depending on the selected **Source**.
 - c. Optionally, enter a **Reason Description** for canceling the policy.
 - d. Select a refund method.

A refund method may default depending on the source and reason.
 - e. Select a different date or accept the default cancellation effective date.

The reason also determines the default cancellation effective date. The system defaults to the earliest allowable date, based on the cancellation reason. The regulations for each jurisdiction govern the earliest allowable date. In most cases, you may add additional days.
- Note:** You can configure this in `CancellationEnhancement.gsx` and in the `notificationconfigs.xml` system table. See “Calculating the Cancellation Effective Date” on page 578 in the *Configuration Guide*.
3. Select **Start Cancellation**.

The **Start Cancellation** button calls `CancellationWizard.pcf`.
4. On the **Confirmation** screen, select **Bind Options** → **Cancel Now** or **Schedule Cancellation**.
 - If you select **Cancel Now**, PolicyCenter sets the cancellation process date to the current date, and soon completes the cancellation after which the cancellation cannot be rescinded. PolicyCenter begins the cancellation process immediately. PolicyCenter sends a notification to the document production system to prepare and mail the appropriate notifications.

To undo the cancellation, you must reinstate the policy.
 - If you select **Schedule Cancellation**, the cancellation process date defaults to the cancellation effective date, which is set by the system. See the table in “Cancellation Overview” on page 108. If you select **Schedule Cancellation**, then PolicyCenter sends a notification to the document production system which prepares and mails the appropriate notifications. PolicyCenter schedules the cancellation, but you have the option to rescind it before the cancellation effective date.

The workflow waits until the system time is equivalent to the `CancelProcessDate` before the cancellation completes. For example, if you select **Insured** as the source with a reason **Out of business**, then the default date is one day after the date you started the cancellation. What happens is that the `CancelProcessDate` is set to the **Cancellation Effective Date**. You can see the details in `JobWizardToolbarButtonSet.Cancellation`. If the policy cancellation is scheduled but has not gone past its cancellation process date, you can rescind the policy as described in the next section.

Rescinding a Cancellation

Rescinding a cancellation on a policy changes the current status of the cancellation to rescinded, and the policy remains in force. PolicyCenter keeps a record of this activity so that you can see rescinded cancellations. Follow these steps to rescind a cancellation:

1. Find a policy with a cancellation pending.

Policies awaiting cancellation display **This policy is pending cancellation** at the top of the screen.
2. From the **Actions** menu, select **Rescind Cancellation** then select the cancellation.

The **Confirmation** screen appears and the status is **Canceling**.

3. Select Rescind Cancellation.

The policy remains in force. You can view the rescinded cancellation in the **Policy Transactions** screen.

Changing the Cancellation Effective Date on an Open Cancellation

If you have the **Cancellation reschedule** permission, you can change the cancellation effective date in an open cancellation.

To change the cancellation effective date in a quoted cancellation

1. Follow the steps in “Canceling a Policy” on page 112 until you arrive at the **Confirmation** screen.
2. Click **Back** to return to the **Entry** screen.
3. Click **Edit Policy Transaction**.

You can now edit the **Reason Description** and **Cancellation Effective Date**.

To change the cancellation effective date in a scheduled cancellation

1. Find a policy with a cancellation pending.
Policies awaiting cancellation display **This policy is pending cancellation** at the top of the screen.
The policy **Summary** screen, displays **Pending Policy Transactions**.
2. Click the **Transaction #** link for a cancellation.
3. Click **Edit Policy Transaction**.

On the **Entry** screen, you can now edit the **Reason Description** and **Cancellation Effective Date**.

Viewing Cancellations in Policy Transactions

You can view the cancellations on a policy in the **Policy Transactions** screen. This screen displays:

- **Effective Date**
- **Transaction Status**
- **Current Status**

Select a cancellation to view cancellation details.

Policy Change Transaction

A policy change transaction is a modification made to a policy while it is in-force. What sort of changes might happen to a policy mid-term?

Common personal auto examples include:

- Adding another person to your policy as a driver
- Increasing your deductible so that you have a less expensive premium
- Adding or removing a vehicle and changing some of the vehicle coverages

Common workers' compensation examples include:

- Adding or changing a location
- Updating the number of employees
- Updating the basis

Common businessowners examples include:

- Changing your business location
- Changing the type or amount of coverage

This topic explains describes the policy change transaction and how to work with policy changes in PolicyCenter.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as s.

This topic includes:

- “Policy Change Overview” on page 116
- “Policy Change General Steps” on page 117
- “Policy Change Key Features” on page 117
- “Working with Policy Changes” on page 122

See also

- “Configuring Policy Change” on page 581 in the *Configuration Guide*

Policy Change Overview

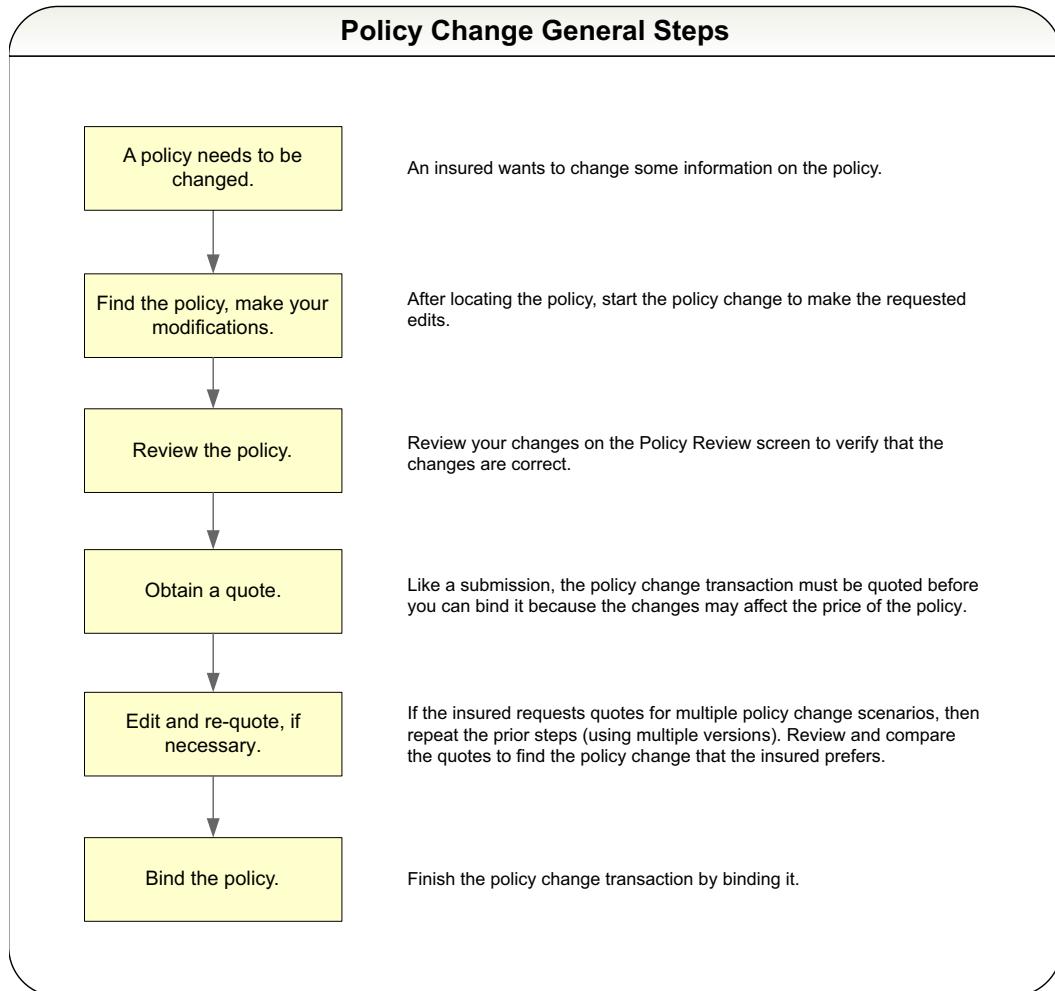
The main purpose of a policy change is to modify one or more elements of a policy. For example, you can use a policy change to change a coverage, exposure, or location. You can add a driver or change the terms of payment. Policy changes occur fairly regularly.

For example, three months into the policy period, the insured contacts the producer to add a second vehicle to the insured’s personal auto policy. The producer finds the policy, makes the requested changes to it, requotes the policy, and then binds it. Two months later, the producer receives another call from the insured to have another family member added to the policy as a second driver on the first vehicle. Again, the producer makes changes. This type of example represents the majority of change policy transactions in PolicyCenter: adding, removing, or changing coverages and coverables, or changing coverage terms. These changes usually have an impact on the premium.

There are other types of less commonly used policy changes, such as out-of-sequence policy changes and preemption. See “Handling Out-of-sequence Policy Transactions in a Policy Change” on page 122 and “Using Preemption in a Policy Change” on page 122 for more information.

Policy Change General Steps

The following diagram shows the basic steps to create a policy change. These steps are for the default configuration of PolicyCenter. Your business requirements and even lines of business can alter the process. The steps may differ slightly between personal and commercial lines and in how you access policy change screens.



Policy Change Key Features

Some of the key features of a policy change transaction in PolicyCenter include:

- “Using the Policy Review Screen to Verify Changes” on page 118
- “Changing the Producer of Service in a Policy Change” on page 118
- “Changing the Policy Expiration Date in a Policy Change” on page 118
- “Editing the Effective Date of a Policy Change” on page 118
- “Handling Out-of-sequence Policy Transactions in a Policy Change” on page 122
- “Using Preemption in a Policy Change” on page 122

See also

- “Multi-version Quoting” on page 169

Using the Policy Review Screen to Verify Changes

In PolicyCenter, you can see your policy changes prior to binding, by viewing the **Policy Review** screen. Use this screen to review your changes and to verify that there are no conflicts. The contents of this screen are dynamic, and it displays:

- Any changes you have made.
- Any conflicts that you need to resolve before binding (if the policy change is out-of-sequence)
- Any conflicts related to preemption.

Note: Although the **Policy Review** screen displays changes to the policy, also review the **Quote** screen to see how your changes may result in a different premium. In the **Quote** screen:

- The **Cost Change Detail** tab displays the transaction cost (offsets and onsets) resulting from the policy change.
- The **Policy Premium** tab displays the breakdown of the premium for the entire policy period.

Changing the Producer of Service in a Policy Change

You can change the Producer of Service or the Producer Code of Service from the **Policy Info** screen.

Note: You cannot change the Producer of Record in a policy change. The Producer of Record can only be changed in a rewrite or renewal.

Changing the Policy Expiration Date in a Policy Change

If you need to change the policy expiration date, navigate to the **Policy Info** screen in the **Policy Details** section. Select **Other** from the **Term Type** drop-down menu to make the **Expiration Date** field editable.

Editing the Effective Date of a Policy Change

You can edit the effective date of an unbound policy change by selecting **Actions** → **Edit** → **Effective Date**. However, after the policy is bound and issued, you can no longer edit the effective date.

Editing the effective date of a policy change is often useful in personal lines of business. A policyholder calls to make a policy change. The agent starts a policy change, enters new policy information and generates a quote. Later, the customer calls requesting a change to the effective date of the policy change.

For example, in a personal auto policy, the policyholder may not know exactly the effective date of a policy change for the purchase of a new vehicle. The policyholder calls the insurance company to determine the cost of coverage for the new vehicle. The agent starts a policy change but does not complete it. The policyholder initially expects to receive the vehicle on a certain date (February 1), so the agent enter February 1 as the effective date of policy change. However, the vehicle actually arrives two weeks late (February 15).

The agent accesses the policy change and changes the effective date. The agent can also make other changes to the policy such as adding additional coverages. The agent quotes the policy. If the policyholder is satisfied, the agent binds and issues the policy.

Note: When you change the edit effective date in a policy change, everything that existed on the policy prior to that change will be on the policy after that change. In your customization, there may be cases where you need to modify or remove what is on the policy. In particular, underwriting issues might have applied on the earlier effective date but are no longer applicable on the new effective date. For an example, “Rule Sets, Gosu Code, and Editing the Effective Date” on page 121.

New Effective Date Must be Within the Same Slice and Policy Term

When editing the effective date of a policy change, the new effective date must not cross any slice boundaries. That is, the new effective date must not cross over the effective date of any other transactions. It must also be within the same policy term.

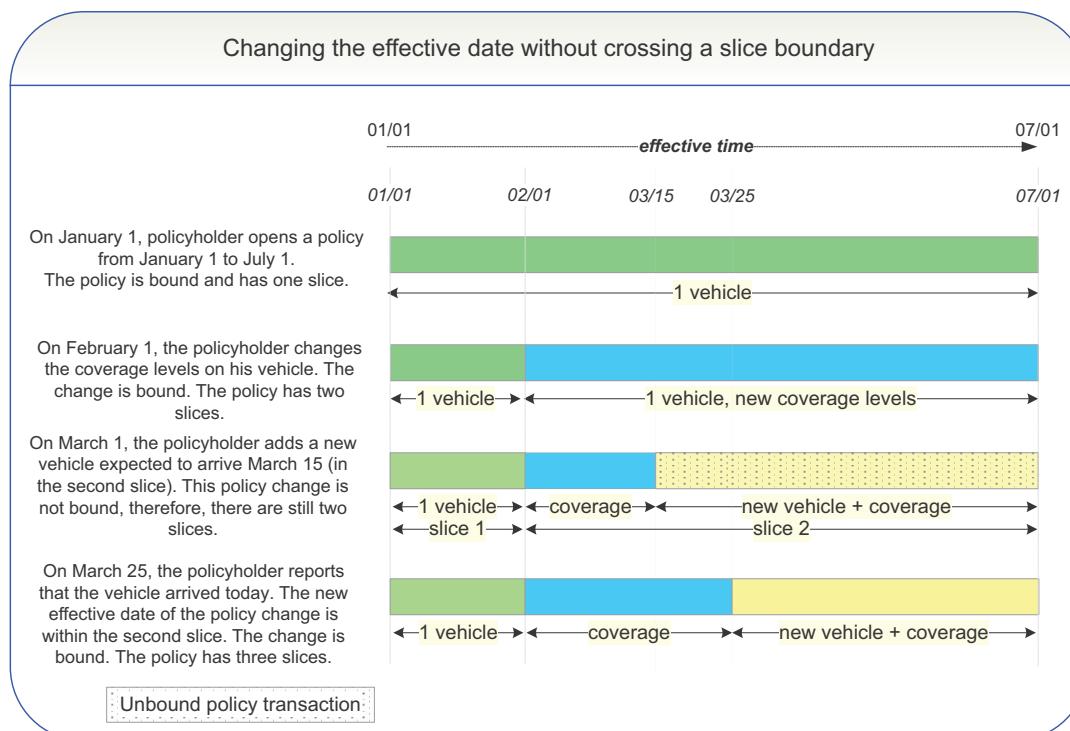
For more information about slices, see “Slice Mode and Window Mode Overview” on page 489.

Example that does not cross slice boundaries

The following example of a change to the effective date does not cross slice boundaries.

1. The agent creates a new personal auto policy with one vehicle. The policy has a six month term from January 1 until July 1. The policy has one slice from January 1 until July 1.
2. On February 1, the policyholder calls to change the coverage levels on his vehicle. The effective date of the policy change is February 1. The agent binds the policy change. The policy now has two slices. The first slice is from January 1 to January 31. The second slice is from February 1 to July 1.
3. On March 1, the policyholder calls to say that he is buying a new vehicle. He expects to receive the vehicle on March 15 (in the second slice). He wants to know the cost of adding the vehicle to his policy. The agent starts a policy change with the expected delivery date. The agent gives the policyholder a quote. The agent tells the policyholder to call with the vehicle identification number (VIN) once he has actually received the vehicle. The VIN is required to bind the policy change. The policy change is quoted but not bound. The policy still has two slices.
4. The policyholder calls on March 25, saying that he has received the vehicle. The agent adds the VIN and changes the effective date of the policy change to March 25. This change to the effective date is allowed because the date is within the second slice. The agent binds the change.

The following illustration shows the slice boundaries in this example.

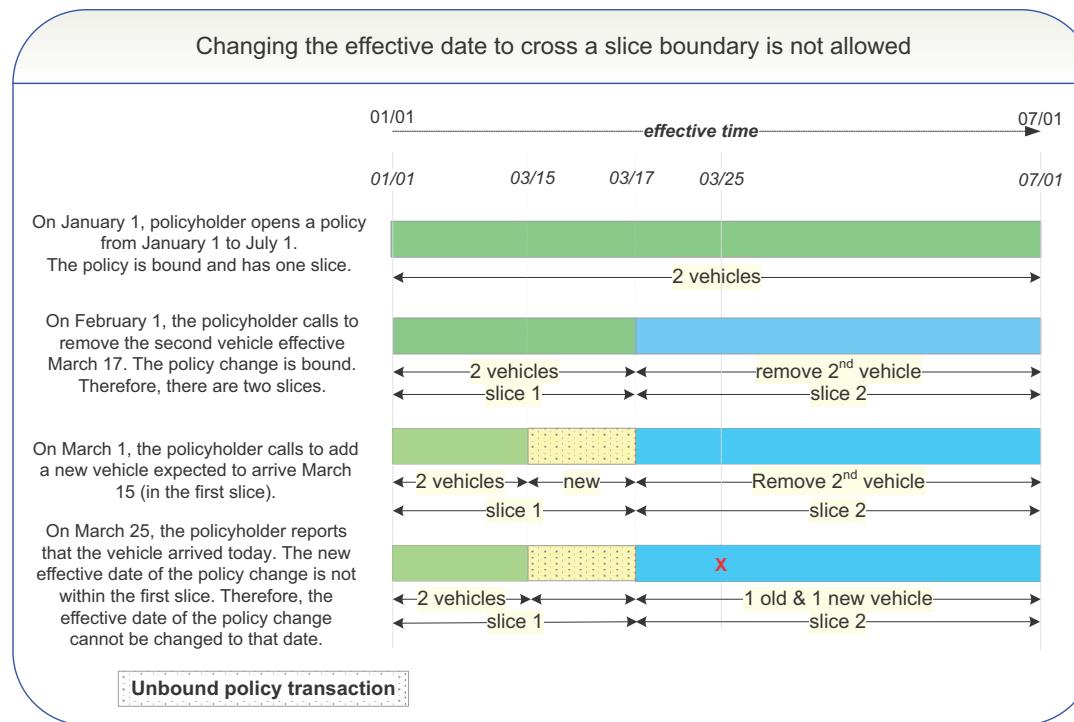


Example that crosses slice boundaries

The following example crosses slice boundaries. Therefore, it is not allowed.

1. The agent creates a new personal auto policy with two vehicles. The policyholder is the primary driver of the first vehicle, and his wife is the primary driver of the second. The policy has a six month term from January 1 until July 1.
2. The policyholder calls on February 1 to report that the lease on the second vehicle expires on March 17. He is not going to renew the lease. The agent starts and binds a policy change. The policy has two slices. The first slice is from January 1 through March 16. The second slice extends from March 17 through July 01.
3. On March 1, the policyholder calls and says that he is buying a new vehicle. He expects the vehicle to arrive on March 15. He wants to know the cost of adding it to his policy. His wife will be the primary driver of the new vehicle.
4. The agent starts a policy change with the expected delivery date. The agent gives the policyholder a quote. The agent tells policyholder to call back with the VIN once he has actually received the car. The VIN is required to bind the policy change. The policy change is within the first slice. The policy change is quoted but not bound. Because the policy change is not bound, the policy still has two slices.
5. The policyholder calls on March 25, and says he has received the vehicle and has the VIN. The effective date of the policy change is not within the first slice. Therefore, the effective date of the policy change cannot be changed to that date. The agent must withdraw the current policy change and start a new one with the same information.

The following illustration shows the slice boundaries in this example.



Multiple Versions

When you select **Actions** → **Edit** → **Effective Date**, the action applies to all versions of a multi-version policy change. PolicyCenter warns you that all versions will have their effective date modified, and all quotes will be invalidated. If any of the versions cannot have the effective date modified for any reason, then no version can have it modified.

See also

- “Multi-version Quoting” on page 169

Underwriting Issues

In general, when you edit the effective date of a policy change, PolicyCenter preserves existing underwriting issues and approvals. This topic describes specific variances of how underwriting issues are handled when you edit the effective date of a policy change.

Approvals Valid for a Specific Amount of Time

In an unbound policy change, a new underwriting issue may be approved for a specific amount of time. If the effective date changes, PolicyCenter recalculates the expiration date of the approval based on the new effective date.

For example, an underwriter approves an issue for three years:

- Policy change effective date: 02/01/2010
- Approval expires: 02/01/2013

The agent changes the effective date, and PolicyCenter adjusts the underwriting approval expiration:

- Policy change effective date: 03/01/2010
- Approval expires: 03/01/2013

Approvals with Expiration Dates

When the user changes the effective date of a policy change, PolicyCenter reevaluates approvals that have an expiration date. For the approval to remain in effect, the new effective date must be before the expiration date of the approval. An approval that had expired on the original date must be reevaluated to determine if it is valid on the new date.

For example, an underwriting issue is approved because the policy change effective date is sooner than the approval expiration date:

- Original policy change effective date: 02/01
- Approval expiration date: 02/05

After the effective date is changed, the approval is no longer in effect:

- New Policy change effective date: 02/08
- Approval expiration date: 02/05

Approvals Invalid from Next Edit

Editing the effective date of a policy change expires an approval if the approval becomes invalid from the next edit.

Rule Sets, Gosu Code, and Editing the Effective Date

In your custom rule sets and Gosu code for handling underwriting issues and approvals, editing the effective date of a policy change might require special handling by that code.

For example, assume the code of an underwriting issue rule creates an underwriting issue if the effective date is before June 1, 2010. An agent creates a policy change effective on May 1, 2010 that triggers that rule to create an underwriting issue. Later, the agent changes the effective date to June 2, 2010. Because the effective date is after June 1, the underwriting issue is now invalid. PolicyCenter automatically keeps that underwriting issue. The rule set or Gosu code needs to handle removing it.

Handling Out-of-sequence Policy Transactions in a Policy Change

Out-of-sequence policy transactions are policy transactions with an effective date is before the effective date of a previous policy transaction on the same policy. Carriers sometimes call these situations *out-of-sequence endorsements*. PolicyCenter uses the term *out-of-sequence*.

A policy can have changes with effective dates that are not sequential. Sometimes this is not an issue. However, there are other times that multiple policy transactions can conflict with each other. PolicyCenter handles these conflicting policy transactions by first recognizing them and second, by allowing you to reconcile any conflicts that result. Policy transactions conflict when a policy change has a transaction date later than another policy transaction, but an effective date earlier than that other policy transaction.

See also

- See “Out-of-sequence Jobs” on page 501

Using Preemption in a Policy Change

A preemption can occur if two policy transactions are open on a policy at the same time. The first policy transaction that binds preempts the second one. PolicyCenter handles preemptions by merging the changes made in the preempting policy transaction with the changes made in the preempted policy transaction. The user interface provides you with a way to merge these changes into the preempted policy transaction. Preemptions are not unique to policy changes; they also occur in audits, cancellations, reinstatements, and renewals.

See also

- See “Preempted Jobs” on page 503

Working with Policy Changes

IMPORTANT Since the policy change process for any carrier can be configured based on business requirements, all discussions apply to the default application except as noted.

This topic explains, from a user’s point of view, how to modify a policy. The ways that you can start a policy change transaction are:

- **Manually through the user interface in PolicyCenter** by selecting **Change Policy** from the **Actions** menu.
- **Externally using the Policy Change API** which allows you to set policy transactions to run manually or automatically. A policy transaction can be started, quoted, and bound automatically. See “Policy Change Web Services and Plugins” on page 583 in the *Configuration Guide* for more information.

With any policy change transaction, certain steps occur:

- Policy changes must be quoted before being bound and are subject to validation at the quotable level.
- Validation is run at the bindable validation level when attempting to bind the change. If validation fails with errors, the process stops and stays in the previously quoted status. If validation fails with warnings, then PolicyCenter stops the first time, but you can override the warnings by clicking **Bind** again.
- If the policy can be bound, then billing instructions are sent to an external billing system through the **IBillingSystemPlugin**.

See also

- “Policy Tools Menu” on page 307
- “Creating and Comparing Multi-version Quotes” on page 170

Creating a Policy Change

1. Navigate to a policy and select **Change Policy** from the **Actions** menu.

2. Specify an effective date and optionally enter a description for the policy change.

Note: Although it is optional, it is useful to provide a description for the policy change. Otherwise, the policy change can only be identified by its effective date. If a policy has more than one policy change, it can be difficult to find a particular policy change if effective date is the only way to distinguish them. This situation is particularly a problem when multiple policy changes have the same effective date.

3. Make changes to the policy contract by using the policy change wizard links on the left of the screen.

4. Review changes made to the policy by selecting **Policy Review**.

This review identifies any additions, removals, or changes to the policy.

5. Since a change typically involves modifications to the exposures or coverages, you must select **Quote** to requote the policy.

The tabs on the **Quote** screen provide different views of the financial impact of the policy change. View the different cards on the **Quote** screen to see the financial impact of the policy change.

- Viewing the **Policy Premium** card itemizes the entire policy as it stands after the change. This information does not distinguish between exposures and coverages previously on the policy and those just being added to (or removed from) the policy.

- The **Cost Change Detail** card shows the transaction cost (offset and onset) resulting from the policy change.

The basic purpose of these cards is to answer various questions an insured might ask when exploring the financial impact of a policy change. The insured might ask, “How much will this cost me right now?” and “What is the total cost for this policy going to be?”

6. After the policy change has been successfully quoted, you can:

- Edit and requote. To edit policy changes, you must have the `viewpolchange` and `editpolchange` system permissions.
- Create a new version.
- Save the draft.
- Select close options.

7. Click **Issue Policy** to bind and issue the policy.

This action is similar to binding a submission. After binding, the change becomes a legal part of the contract as of the change’s effective date.

Editing the Policy Change Effective Date

You can edit the policy change effective date in an unbound policy.

1. Find a policy change that has not been bound and issued.

2. Select **Actions** → **Edit** → **Effective Date**.

The **Policy Change Summary** screen appears.

3. Modify the **Effective Date** and, optionally, the **Description**.

The effective date must be:

- Different than the current effective date.
- Within the current slice. See “New Effective Date Must be Within the Same Slice and Policy Term” on page 119.

4. Click **Next** to advance to other screens in the policy change wizard.

You can make other changes to the policy.

5. Click **Quote or Save Draft**.
6. Click **Issue Policy** if you wish to bind and issue the policy change. After you issue the policy, you are no longer able to edit the effective date of the policy change.

Reinstatement Policy Transactions

Reinstatements go hand in hand with cancellations and are a type of policy change that returns a canceled policy to in force status. The policy becomes in force again as of the reinstatement date. The reinstatement removes the cancellation from the policy period. Hence, the policy expiration date remains the same.

In a reinstatement policy transaction, you cannot reinstate with a lapse in coverage or change the policy expiration date. To reinstate with a lapse in coverage, you must do a rewrite policy transaction. For more information, see “Rewrite Policy Transactions” on page 129.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

This topic includes:

- “Reinstatement Overview” on page 125
- “Reinstatement General Steps” on page 126
- “Reinstatement Key Features” on page 126
- “Working with Reinstatements” on page 126

See also

- “Configuring Reinstatement” on page 585 in the *Configuration Guide*

Reinstatement Overview

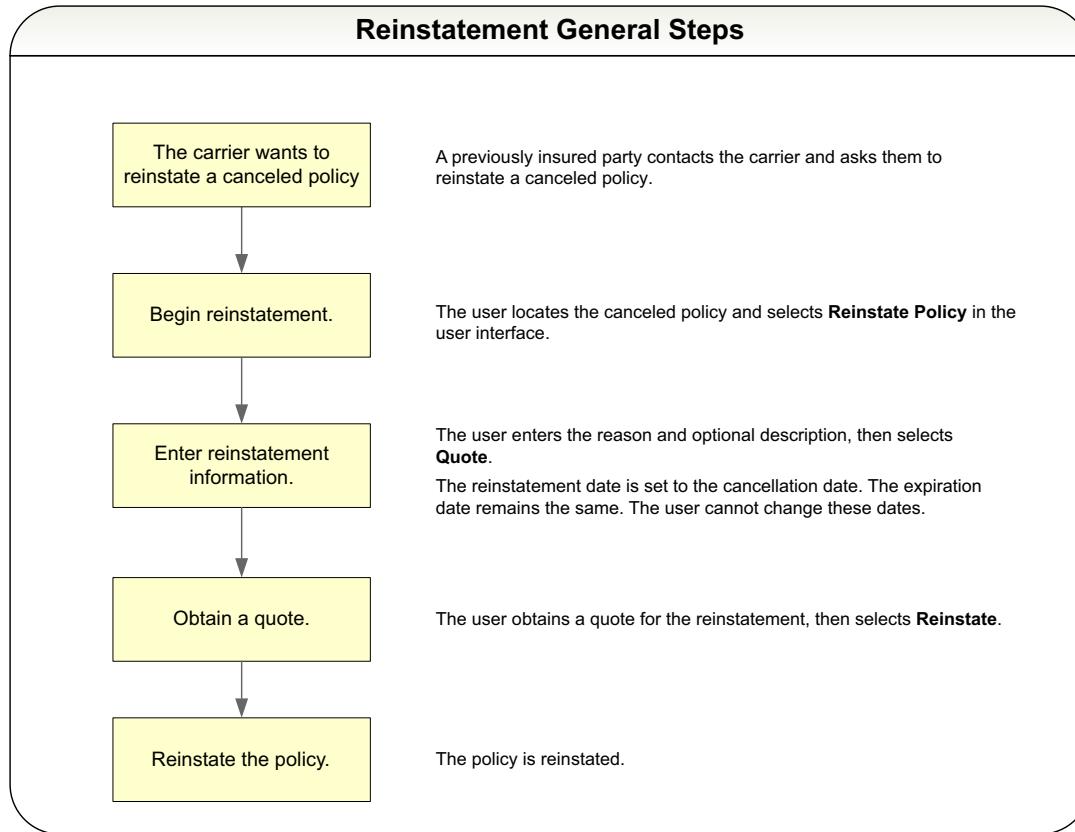
Note: Since the reinstatement process for any carrier can be configured based on business requirements, all discussions apply only to the base configuration.

A reinstatement policy transaction can be started either manually in the PolicyCenter Policy File or programmatically.

From the policyholder's perspective, a reinstated policy is no different than the original policy. However, PolicyCenter tracks the reinstatement as a policy transaction.

Reinstatement General Steps

The following diagram shows the basic steps in a reinstatement. These steps are for the default configuration of PolicyCenter. You business requirements and even lines of business can alter the process. Steps may also differ slightly between personal and commercial lines and in how you access the reinstatement screens.



Reinstatement Key Features

Reinstatement provides the following key features:

- You can reinstate a canceled policy.
- You cannot reinstate a canceled policy with a new expiration date.
- The reinstated policy cannot have a lapse in coverage.

Note: You cannot make any changes to the policy in a reinstatement.

Working with Reinstatements

This topic describes how to work with reinstatements in the user interface.

See also

- “Policy Tools Menu” on page 307

Reinstating a Policy

This topic explains, from a user’s point of view, how to reinstate a policy.

Note: The following describes a default reinstatement policy transaction. Your configuration and line of business will affect the look and flow.

Reinstatements must start from a canceled policy.

1. Find a canceled policy and select **Reinstate Policy**.

The Reinstatement wizard displays the **Start Reinstatement** screen with the **Effective Date of Reinstatement** set to the **Cancellation Effective Date**.

2. Enter a **Reason**, an optional **Reason Description**, and select **Quote**.

The Reinstatement wizard begins, and the **Quote** screen appears.

3. From the **Quote** screen, you can either edit, reinstate, withdraw the policy, or print a quote.

- a. Selecting **Edit** allows you to only edit the **Reason Description** field. You must select **Quote** to return it to the status of quoted. However, the quote does not change.
- b. Selecting **Reinstate** reinstates the policy and you are able to either return to the policy file or your desktop.
- c. Selecting **Withdraw Transaction** stops the process and the policy remains in the status of canceled. However, you can decide at a later date to reinstate the policy.
- d. Selecting **Print Quote** prints the quote for review.

Rewrite Policy Transactions

Carriers must do a rewrite if they need to change the effective date of the policy or producer of record. Carriers cannot change these in a policy change transaction.

Note: In a policy change, you can change the producer of service but not the producer of record. In a policy change, you can modify policy information but the effective date must remain the same.

Carriers may choose to rewrite a policy when the policy has errors or significant changes. For example, the producer reviews the policy documentation before it is sent out and notices that the name of the insured is misspelled. The carrier can use a policy change to correct the name, however the name will be corrected in an addendum to the policy but not in the policy itself. The carrier decides to do a full-term rewrite of the policy, which reissues the documentation with the insured's name spelled correctly. The policy rewrite is transparent to the insured because the insured never receives the original policy and documentation.

Although significant changes to a policy can be done as a policy change, it may be preferable to do these as a rewrite. For example, the insured calls and asks that the billing method be changed from agency to direct. The carrier makes this change as a mid-term rewrite to simplify tracking of this change for both the carrier and the insured. Rewrite reissues the policy documentation rather than sending an addendum, and the carrier creates a completely new policy. Rewrite makes it easy to keep track of when the change occurred.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

This topic includes:

- “Rewrite Overview” on page 130
- “Rewrite General Steps” on page 131
- “Rewrite Key Features” on page 131
- “Working with Rewrites” on page 132

See also

- “Configuring Rewrite” on page 589 in the *Configuration Guide*

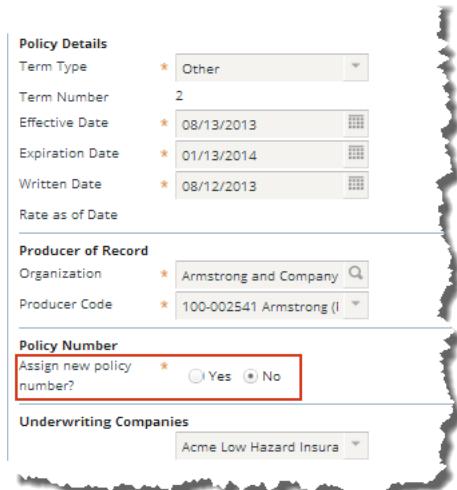
Rewrite Overview

Note: Since the rewrite process for any carrier can be configured based on business requirements, all discussions apply only to the out-of-the-box application.

The rewrite policy transaction allows a user to completely rewrite a policy. It creates a new policy version that can still be tracked to the original submission. Rewrite does not appear as a menu option until a policy has been canceled. The user must first manually cancel the policy.

Rewrite policy transactions are similar to Submission policy transactions. However, rewrites can be configured independently, so they have separate user interface screens, wizard flow, permissions, and rules.

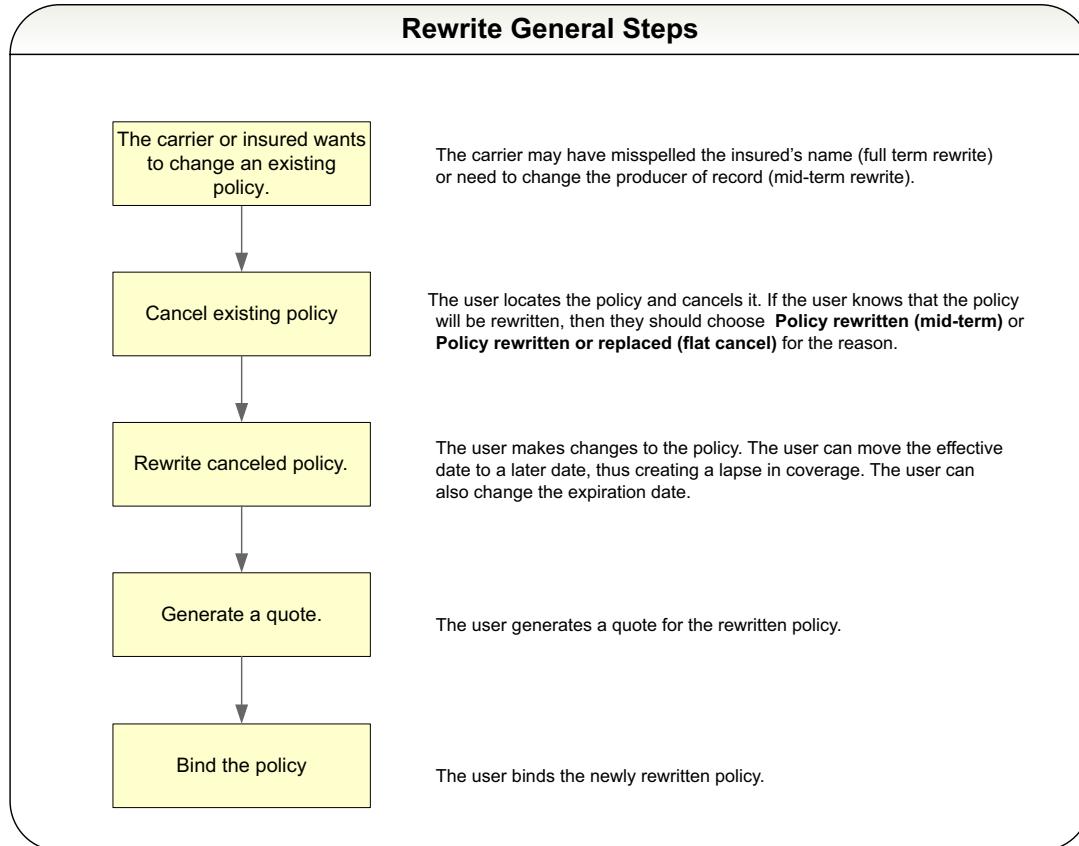
There are a few differences between the **Policy Info** screen in the Rewrite wizard and the Submission wizard. You can change any or all of the policy information details. However, Rewrite has a Boolean radio button which allows you to assign a new policy number. If selected, then PolicyCenter assigns a new policy number when binding the rewrite; otherwise the policy number remains the same. So from the policy holder's perspective, they have received a completely new policy, and in PolicyCenter, both the newly rewritten policy and the original policy version still exist. The following screen shot shows the **Policy Details** section of the **Policy Info** screen in the rewrite policy transaction.



The screenshot shows the 'Policy Details' section of a web-based application. It includes fields for Term Type (set to 'Other'), Term Number (set to '2'), Effective Date ('08/13/2013'), Expiration Date ('01/13/2014'), Written Date ('08/12/2013'), and Rate as of Date. Below this is the 'Producer of Record' section with Organization ('Armstrong and Company') and Producer Code ('100-002541 Armstrong (I)'). The 'Policy Number' section contains a question 'Assign new policy' with radio buttons for 'Yes' (selected) and 'No'. At the bottom is the 'Underwriting Companies' section with a dropdown menu showing 'Acme Low Hazard Insura'.

Rewrite General Steps

The following diagram shows the basic steps in a rewrite. These steps are for the default configuration of PolicyCenter. Your business requirements and even lines of business can alter the process. For example, the steps may differ slightly between personal and commercial lines and in how you access rewrite screens.



Rewrite Key Features

The rewrite feature allows you to do the following:

- “Change Any Policy Information” on page 131
- “Full-Term Rewrite” on page 131
- “Mid-Term Rewrite” on page 132

Change Any Policy Information

In a rewrite you can change anything in a policy. In particular, rewrite allows you to change the producer of record and the policy effective and expiration dates. (In a policy change, you can change the producer of service but not the producer of record.)

Full-Term Rewrite

A full-term rewrite replaces the original policy for the complete policy term. A full-term rewrite can have a lapse in coverage. See “Rewriting a Policy with a Lapse in Coverage” on page 132.

Mid-Term Rewrite

A mid-term rewrite replaces a portion of the original term and allows you to rewrite the policy to the original policy end date or to a new end date. A mid-term rewrite can create a lapse in coverage. See “Rewriting a Policy with a Lapse in Coverage” on page 132.

Working with Rewrites

This topic explains, from a user’s point of view, how to rewrite a policy.

- “Rewriting a Policy” on page 132
- “Rewriting a Policy with a Lapse in Coverage” on page 132
- “Approving, Requesting Changes, or Declining a Rewrite” on page 132
- “Viewing Rewritten Policies” on page 133

IMPORTANT The following describes a default rewrite policy transaction, therefore, the following screen shots serve as an example. Your configuration and line of business can affect the look and flow.

See also

- “Policy Tools Menu” on page 307

Rewriting a Policy

The rewrite process starts from a canceled policy. For instructions on how to cancel a policy, see “Working with Cancellations” on page 111.

1. Select the canceled policy.

The policy **Summary** screen appears.

2. From **Actions** menu, select **Rewrite Full Term**, **Rewrite Remainder of Term** or **Rewrite New Term**.

If the cancellation reason was **Policy rewritten or replaced (flat cancel)**, only **Rewrite Full Term** is available. For other cancellations **Rewrite Remainder of Term** and **Rewrite New Term** are available.

If you select **Rewrite Remainder of Term**, you can make changes to the policy including the **Effective Date** and **Expiration Date**. By default, **Effective Date** is set to the cancellation date. You can change **Effective Date** to a later date but not an earlier date. Changing to a later date creates a lapse in coverage.

If you select **Rewrite New Term**, you can make changes to the policy including **Term Type** and **Effective Date**. By default, **Effective Date** is set to the cancellation date and **Expiration Date** is set to **Effective Date** plus the term. You can change **Effective Date** to a later date but not an earlier date. Changing to a later date creates a lapse in coverage.

3. Make your changes to the policy, and then select **Quote**.

4. Select **Bind** to issue the policy.

Rewriting a Policy with a Lapse in Coverage

To rewrite a policy with a lapse in coverage, simply change the **Effective Date** to a later date (see step 2 of “Rewriting a Policy” on page 132).

Approving, Requesting Changes, or Declining a Rewrite

An underwriter may choose to approve, request changes, or decline (by withdrawing) a rewrite. A user first cancels a policy with the intent to rewrite it. Based on business rules, the user does not have the authority to complete the rewrite because it needs underwriting approval. PolicyCenter sends the rewrite to the underwriter,

and alerts the underwriter by an activity. The underwriter reviews the rewrite and may choose to approve it or edit it then requote the policy. The underwriter selected **Approve Options** on the rewrite toolbar. The underwriter can also decline the rewrite by withdrawing the rewrite policy transaction. PolicyCenter notifies the user of the underwriter's action. The user can contact the policyholder and send the appropriate documentation.

Viewing Rewritten Policies

Rewriting a policy creates a new version of the policy. You can view all versions of the policy by going to the Account and viewing the **Account File Summary** screen. The following illustration shows the original policy that was canceled, and the full-term rewrite of the policy that is currently in force.

The screenshot displays the 'Account File Summary' interface. At the top, there are buttons for Refresh, Edit Account, and Change Account Holder To. Below this, account details are shown: Account No (C000143542), Account Name (Ray Newton), and Status (Active). To the right, an 'Official IDs' section shows an SSN (342-56-87). A decorative graphic of black chevrons is positioned between the account details and the policy terms. Below the chevrons, there are three small icons followed by the word 'Normal'. Next to them is a button labeled 'All ordered MVRs received - Clear'. To the right of this is a dropdown menu set to 'Alice Applegate'. The main content area is titled 'Policy Terms' and contains a table with two rows:

Policy #	Product	Status	Effective Date	Expiration Date
8691790551	Personal Auto	Canceled	07/13/2013	01/13/2014
8691790551	Personal Auto	In Force	08/13/2013	01/13/2014

At the bottom of the table, a link labeled 'Pending Policy Transactions' is visible.

Rewrite New Account Policy Transaction

In PolicyCenter, you can move a policy going forward to a new target account. The previous policy terms remain on the initial account. For example, a young adult has a policy in his parent's personal auto account. He graduates from college, and wants to move his policy to his own account. The insurer cancels his policy, and rewrites it to his new account.

When you rewrite policies to a new account, PolicyCenter creates a rewrite new account policy transaction for each policy. This policy transaction takes data from an existing policy and creates a new policy with a new policy number in the new account. You can only rewrite canceled or expired policies to a new account.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

This topic includes:

- “Rewrite New Account Overview” on page 136
- “Working with Rewrite New Account” on page 141

See also

- “Configuring Rewrite New Account” on page 591 in the *Configuration Guide*
- “Overview of Moving and Rewriting Policies Between Accounts” on page 324
- “Rewriting Policies From One Account to Another” on page 332
- “Overview of Split and Spin-off Policies” on page 305

Rewrite New Account Overview

The rewrite new account policy transaction creates a completely new policy, but PolicyCenter treats the source policy and the new policy functionally as one policy. Therefore, this policy transaction enforces the business requirement that, even though the source and rewritten policy are on different accounts, the active policy periods may not overlap. This business requirement is enforced throughout the life of both policies. If necessary, the active policy periods can have a gap between them.

The rewrite new account policy transaction has similarities to both submission and a rewrite policy transactions.

Similarities to submission

- Results in a new policy with a new policy number.
- Provides a qualification step.
- Provides billing similar to a submission.

Similarities to rewrite

- Is based on an existing policy period.
- Effective dates cannot overlap with the policy period it is based on.
- May result in out-of-sequence conflicts. If the based-on policy has future slices, they are rewritten to the new policy. The start of the rewrite new account policy transaction may be out of sequence in relation to these future slices.

Restrictions to Policy Transactions Started on the Source Policy

The rewrite new account policy transaction rewrites a source policy to a target policy. There are a number of restrictions that apply to a rewrite new account policy transaction. Primarily, the restrictions prevent the target policy from having an active term that overlaps with the source policy.

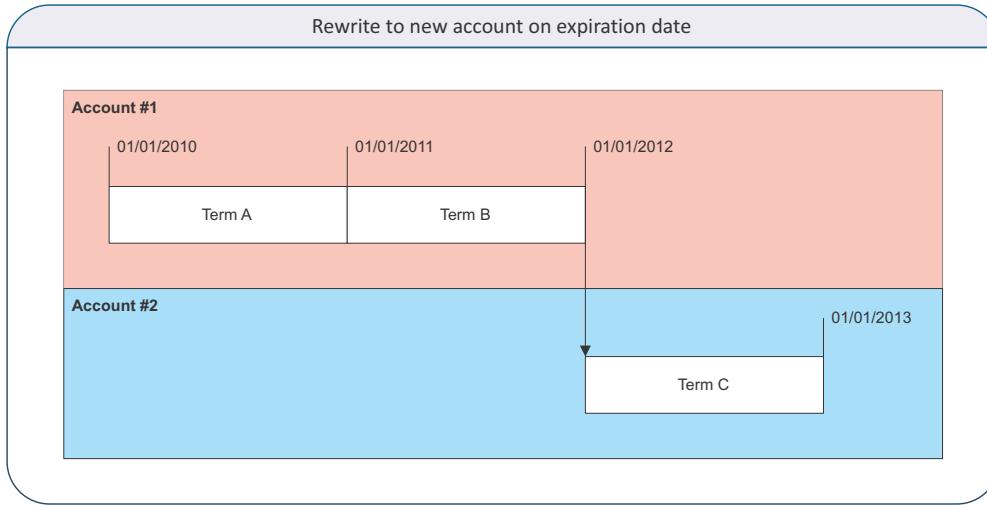
Note: While the rewrite new account policy transaction is open, no policy transactions other than audit can be started on the source policy.

- “Rewrite to New Account On or After Expiration Date” on page 136
- “Rewrite to New Account on Cancellation Date” on page 137
- “Rewrite to New Account on Future Date Within the Policy Term” on page 138
- “Rewrite to New Account on a Future Date Beyond the Policy Term” on page 139

Rewrite to New Account On or After Expiration Date

When you rewrite a policy to a new account, the effective date of the policy can occur on or after the expiration date of the source policy. Policy transactions on the source policy have some restrictions.

In the illustration below, Account #1 is the source policy with an expiration date of 01/01/2012. Account #2 is the rewritten policy with an effective date of 01/01/2012. The effective date of Account #2 can be a date on or after 01/01/2012.



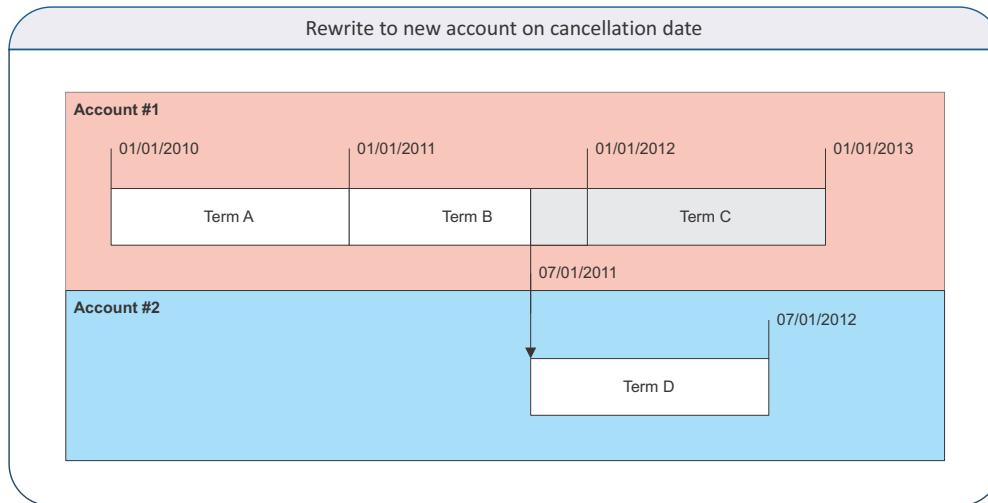
The following table shows restrictions for policy transactions on the source policy.

Policy transaction	Restriction
Submission	Not applicable.
Policy Change	Not allowed on and after the expiration date of the source policy.
Cancellation	Not allowed on and after the expiration date of the source policy. Does not differ from the usual behavior.
Reinstatement	The end of the term being reinstated cannot be after the cancellation date of the source policy.
Rewrite	Not allowed on and after the expiration date of the source policy. Therefore, you cannot rewrite this policy.
Renewal	Not allowed on or after the expiration date of the source policy.
Audit	Allowed.
Rewrite New Account	Not allowed.

Rewrite to New Account on Cancellation Date

When you rewrite a policy to a new account, the effective date of the policy can be the cancellation date of the source policy. Policy transactions on the source policy have some restrictions.

In the illustration below, Account #1 is the source policy with a cancellation date of 07/01/2011. Account #2 is the rewritten policy with an effective date of 07/01/2011.



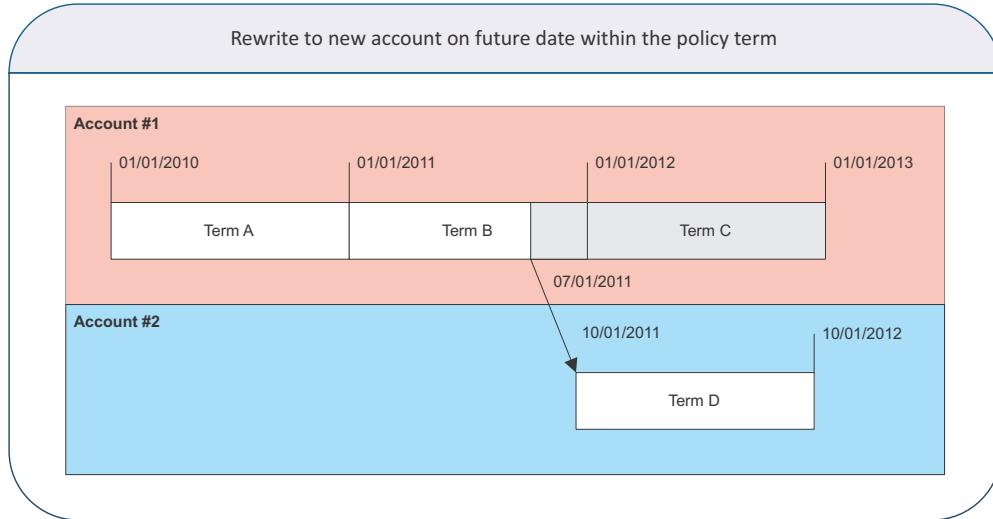
The following table shows restrictions for policy transactions on the source policy.

Policy transaction	Restriction
Submission	Not applicable.
Policy Change	Not allowed on and after the cancellation date of the source policy.
Cancellation	Not allowed on and after the cancellation date of the source policy. Does not differ from the usual behavior.
Reinstatement	The end of the term being reinstated cannot be after the cancellation date of the source policy.
Rewrite	Not allowed on and after the cancellation date of the source policy. Therefore, you cannot rewrite this policy.
Renewal	Not allowed on or after the cancellation date of the source policy.
Audit	Allowed.
Rewrite New Account	Not allowed.

Rewrite to New Account on Future Date Within the Policy Term

When you rewrite a policy to a new account, the effective date of the policy can be a future date within the canceled policy term. Policy transactions on the source policy have some restrictions.

In the illustration below, Account #1 is the source policy with a cancellation date of 07/01/2011. Account #2 is the rewritten policy with an effective date of 10/01/2011.



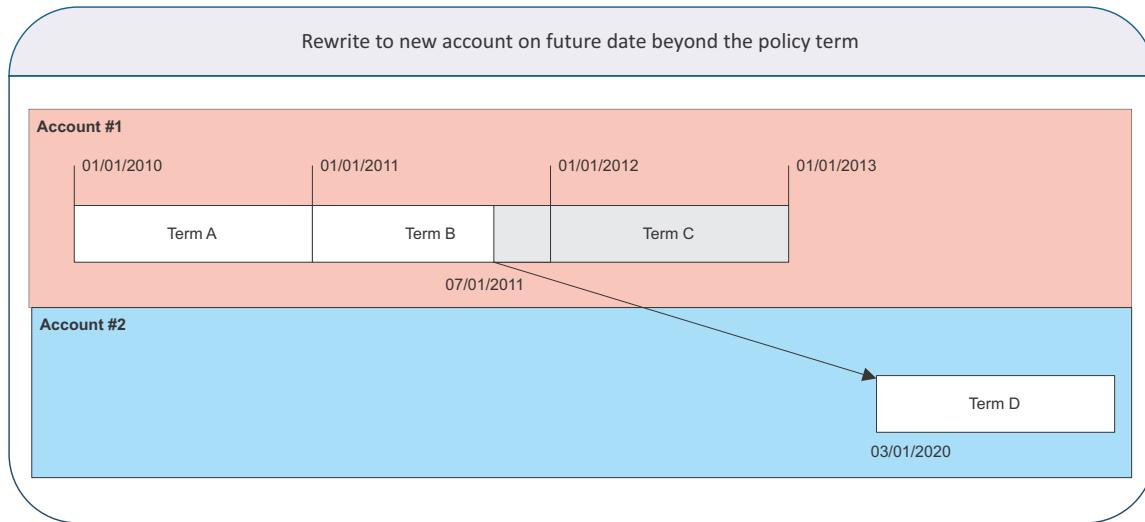
The following table shows restrictions for policy transactions on the source policy.

Policy transaction	Restriction
Submission	Not applicable.
Policy Change	Not allowed on and after the effective date of the rewritten policy (Term D).
Cancellation	Not allowed on and after the cancellation date of the source policy. Does not differ from the usual behavior.
Reinstatement	Not allowed on and after the cancellation date of the source policy because the reinstated period would overlap the rewritten policy (Term D). To reinstate the gap from 07/01/2011 until 10/01/2011: <ol style="list-style-type: none">1. Do a policy change on term B, and set the period end date to 10/01/2011, the effective date of the rewritten policy.2. Reinstate the policy.
Rewrite	Not allowed on and after the cancellation date of the source policy because the rewritten policy overlaps Term D or its successors. To rewrite the gap from 07/01/2011 until 10/01/2011: <ol style="list-style-type: none">1. Do a policy change. Change the end date of Term B to 10/01/2011, the effective date of the rewritten policy.2. Rewrite the remainder of the term.
Renewal	Not allowed on or after the effective date of the rewritten policy (Term D). Therefore, Term C cannot be renewed.
Audit	Allowed.
Rewrite New Account	Not allowed.

Rewrite to New Account on a Future Date Beyond the Policy Term

When you rewrite a policy to a new account, the effective date of the policy can be a future date beyond the policy term and any pending renewals. Policy transactions on the source policy have some restrictions.

In the illustration below, Account #1 is the source policy with a cancellation date of 07/01/2011. Account #2 is the rewritten policy with an effective date of 03/01/2020.



The following table shows restrictions for policy transactions on the source policy.

Policy transaction	Restriction
Submission	Not applicable.
Policy Change	Not allowed on and after the effective date of the rewritten policy (Term D).
Cancellation	Not allowed on and after the cancellation date of the source policy. Does not differ from the usual behavior.
Reinstatement	Not allowed on and after the start date of the rewritten policy (Term D), because the reinstated period would overlap the rewritten policy. In addition, a canceled period that overlaps the start date of the rewritten policy cannot be reinstated.
Rewrite	Not allowed on and after the start date of the rewritten policy because the rewritten policy overlaps Term D or its successors. Rewrite is allowed if the expiration date does not overlap term D.
Renewal	Not allowed on or after the effective date of the rewritten policy (Term D). Term C can be renewed as long as the expiration date does not overlap the effective date of Term D.
Audit	Allowed.
Rewrite New Account	Not allowed.

Restrictions to Policy Transactions Started on the Target Policy

A rewrite policy transaction on the target policy cannot set an effective date that overlaps with the expiration or cancellation date of the source policy.

Other policy transactions do not allow you to move the effective date of a policy to an earlier date. Therefore, this restriction does not affect these policy transactions.

Rewrite New Account and the Renewal Batch Process

The Policy Renewal Start batch process does not process an expired policy that has been rewritten to a new account. The expired policy is not processed because the renewed source policy could create an active period that overlaps the effective dates of the rewritten policy.

Working with Rewrite New Account

This topic provides step-by-step instructions on how to rewrite policies to a new account in PolicyCenter.

Rewriting a Policy to a New Account

1. Rewrite policies from one account to another by following the steps in “Rewriting Policies From One Account to Another” on page 332. When you rewrite policies from one account to another, PolicyCenter does the following:

- Starts a rewrite new account policy transaction for each policy.
- Creates an activity for each new rewrite new account policy transaction and assigns it to the current user.
- Creates a history event on the policy term of the source period for each rewrite new account policy transaction.

2. As the same user, go to the **Account File Summary** screen.

3. In **Current Activities**, click the **Rewrite to new account** activity generated in step 1.

PolicyCenter jumps to the **Rewrite New Account** policy transaction. The bottom part of the window displays the **Activity Detail** tab.

4. Click through the wizard steps making changes as necessary.

On the **Policy Info** screen, the default **Effective Date** is the cancellation or expiration date of the source policy. You can change the **Effective Date** as long as the policy period does not overlap with the source policy period.

5. Quote and issue the policy.

6. In the **Activity** tab at the bottom of the screen, click **Complete** to complete the activity.

Premium Audit Policy Transaction

PolicyCenter supports two types of premium audits: *final audit* and *premium report* policy transactions. A final audit policy transaction covers the entire policy term. Only one final audit applies for each policy term. The final audit begins on the policy effective date and ends on the policy expiration or cancellation date. Premium report policy transactions, on the other hand, are a series of non-overlapping periodic audits that are scheduled and billed within the coverage period. They are also known as interim reports. For example, you can choose to schedule premium reports by calendar months. Then a separate audit is conducted for each calendar month of the policy term.

A final audit contains the verified and ultimate cost for a variable basis policy. When the policy is issued, the estimated annual premium (EAP) is based on the policyholder's best guess at the basis, such as payroll, for the entire policy year. The final audit is conducted at expiration or cancellation. A premium auditor reviews the policyholder's records, or the policyholder officially reports the actual payroll amounts for the past policy term. The cost of the policy is recalculated using this actual basis amount, and the policyholder is billed or returned the difference.

With premium report policy transactions, the policyholder is billed for premium based on periodic requests for actual basis amounts, such as payroll. A deposit, usually a percentage of the EAP, is billed at the beginning of the policy. As each reporting period ends, the policyholder is billed based on the actual basis reported by them. Take, for example, a policy which runs from January 1, 2009 to January 1, 2010 with monthly premium reporting. The policyholder will be billed a deposit and up to 12 monthly reports will be scheduled. At the end of January, PolicyCenter initiates the first monthly report which covers the month of January. By mid-February, the policyholder sends back the basis detail. The application calculates the premium for the month of January and bills the insured. These reports continue on a specified schedule until the policy ends. A final audit is also conducted. The final audit verifies and adjusts the premium for the entire policy term. It also prompts the return of the initial deposit.

In PolicyCenter, final audit policy transactions are available for both workers' compensation and general liability lines of business. Premium report policy transactions are available for the workers' compensation line of business only.

Note: PolicyCenter contains an integration with Guidewire BillingCenter. This topic describes PolicyCenter when this integration is not enabled. For details on how PolicyCenter integrates with BillingCenter see "Billing System Integration" on page 701.

This topic includes:

- “Final Audit Overview” on page 144
- “Premium Report Overview” on page 146
- “Audit Roles” on page 148
- “Premium Audit and Other Policy Transactions” on page 149
- “Premium Audit General Steps” on page 151
- “Working with Final Audits” on page 152
- “Working With Premium Reports” on page 156

See also

- “Configuring Premium Audit” on page 595 in the *Configuration Guide*

Final Audit Overview

Note: In the base application, the workers’ compensation and general liability lines of business are set as auditible.

Some lines of business will require final audit; other lines may offer an optional audit. When the final audit is optional, the underwriter can set the **Requires Final Audit** field on the **Payment** page to Yes, No, or Determined By Business Rules.

Various criteria determine whether a final audit is required, the audit method, and the audit assignment. These criteria vary by carrier. Among other things, the criteria may include the type of exposures that the policy contains, the premium size of the policy, the jurisdiction of the policy coverage.

Audit Schedules

Audit schedules offer the user choices about the audits to be scheduled. For example, final audit schedules determine the audit method, the process start date, and the due date, of the audit method.

Process Start Date Examples

An *expiration final audit* ends on the normal expiration date of the policy. The audit schedule for an expiration final audit which will be completed by a physical audit may be set to start 30 days prior to the policy expiration.

A *cancellation final audit* ends on the policy cancellation date. The audit schedule, which will be completed by a physical audit, may be set to start as soon as the policy is canceled.

Due Date Examples

It is likely that a physical audit will take a longer time to complete than a phone audit. Therefore, the physical audit schedule may set the due date to 45 days from the period end date. The phone audit may set the due date to only 15 days from the period end date.

Audit Method Examples

The default application provides the following audit methods:

- **Physical** – A representative (such as a premium auditor) makes an in person visit to the policyholder to review the business records and verify and obtain required audit information.
- **Phone** – The premium auditor contacts the policyholder by phone to obtain all audit information.
- **Estimated** – This method is used only when the other methods are not available. For example, the policyholder went out of business, and the records were lost.

- **Voluntary** – The carrier sends a form to the policyholder requesting the required information. The policyholder completes the form and returns it to the carrier for processing.

On the process start date, the **Audit Task** batch process starts the audit policy transaction. The status of the audit changes from *scheduled* to *in progress*, and you can begin collecting the audit information. After the information is received and entered, the rating engine uses these actual values to calculate the premiums for the prior policy term. PolicyCenter adjusts the amount billed for the policy. Final audits are always completed after the policy term ends through expiration or cancellation. For expiration final audits, the audit process starts shortly before the renewal policy term begins and the prior policy term concludes.

Note: Since a final audit policy transaction changes the billing, but not the policy contract, the audit policy transaction is not *bound* but is considered *complete*.

If the final audit is scheduled, but it is later determined that it is not required, it may be *waived*. If a final audit is completed and then the audited values need to be adjusted, the final audit may be *revised*. When policy changes are made after a final audit has been completed, PolicyCenter *reverses* the final audit. PolicyCenter schedules a new audit based on the changed policy so that the policy changes may be incorporated into the new audit.

Where Does the Carrier Send Premium Audit Information?

Final audits are important to determine actual policy premium. The carrier must also report final audit information to regulatory agencies. For example, worker's compensation *unit statistical reports* (unit stat reports) are sent to jurisdictional bureaus or a council that handles multiple jurisdictions (NCCI, for example). The jurisdiction uses unit stat reports to calculate policyholder experience modifications and to calculate suggested or mandatory rates obtained by the carrier. PolicyCenter does not create or send unit stat reports. However, you can configure PolicyCenter to export final audit data for statistical reporting. This data can be imported to another system such as the unit stat software application used by a particular jurisdiction.

Reversing and Revising Final Audits

If you need to make changes to a final audit, you can reverse or revise it.

Reverse a Final Audit

The goal of a reversal is to undo the original audit so that a new audit can incorporate changes made by another policy transaction. In addition, PolicyCenter notifies the billing system that the first audit is reversed and will likely be replaced by an entirely new audit. Reversal occurs as a result of another policy transaction becoming effective within the final audit period after the final audit has been completed. In the base configuration, reversal occurs when PolicyCenter processes a cancellation, policy change, or reinstatement on a policy that has a completed final audit. There is no explicit user action. When an audit is reversed, PolicyCenter calls the billing system to reverse charges related to the final audit and sets the reversal date to the current date. In addition to reversing the final audit, PolicyCenter schedules a new audit that replaces the reversed one. After the audit task initiates the new scheduled audit, users can begin processing the new audit.

Revise a Final Audit

You can use an audit revision to change a completed audit. The existing completed audit is the basis for the revision. The revision allows a user to amend the audit details and recalculate the premiums. PolicyCenter forwards the revised audit value or the change in premium to the billing system. This audit becomes the most current representation of the policy premiums.

Click **Revise** in the user interface to begin revising an audit. This action creates a new audit with an audit type of **Final Audit (revised)**.

When might you revise an audit? A premium auditor completes a final audit with an audited payroll. At some later date, the premium auditor realizes that someone entered the amount incorrectly. The premium auditor revises the original final audit and enters the correct amount.

Final Audits Do Not Alter the Policy Contract

PolicyCenter policy transactions, such as submissions, policy changes, or rewrites, are *bound* when completed. Unlike those policy transactions, a final audit policy transaction is *locked* when completed. The final audit policy transaction does not change the policy contract.

Adding a Final Audit

Usually, PolicyCenter schedules final audits automatically for policies that need them. PolicyCenter allows the user to add a final audit in the following cases:

- If the user waived the final audit, then later decide that they need one.
- The application did not schedule a final audit, but the user decides that one is needed.
- If the final audit was reversed without scheduling a new one.

A period can only have one final audit scheduled, in progress, or completed. This limit does not include reversed or revised audits.

See also

- “Adding a New Audit” on page 154 for instructions on adding a final audit

Premium Report Overview

Note: In the base application, premium report are configured for the workers’ compensation line of business.

Premium report policy transactions allow the carrier to bill the premium at regular intervals throughout the policy term based on reported values. These billings attempt to ensure that the premium billed is close to the final audit amount. In most cases, the billings are more accurate than an estimate. For direct bill policies, the insured sends payment along with the report. If the insured does not submit reports in a timely fashion, the carrier can cancel the policy.

The insured may choose premium reporting because they are not able to accurately predict their payroll in advance or they have variable bases, such as seasonal variations in their payroll. Others may choose premium reporting because they end up paying only for the premiums that they actually owe. They may prefer this to paying for everything up front or agreeing to an estimated amount.

On binding a new policy period, the insured is billed a collateral amount called a deposit. The application schedules premium reports based on the audit schedule selected. As each report comes in, the user enters the reported amounts and the rating engine calculates the premium. When the user submits the report, PolicyCenter sends the transactions to the billing system. If a payment is received with the report, then the billing system reconciles the premium amount with the amount that the policyholder sent. Since the insured is doing their own calculation of premium outside the system and sending in a payment, there may be discrepancies that the billing department must resolve.

Within a submission, the premium report policy transaction is a payment plan choice on the **Payment** page. If you select premium reporting, then you can select one of the audit schedules configured for premium reporting. The audit schedule determines the frequency and number of premium reports. The **Payment** page includes another field for scheduling a final audit. The choices are **Yes** (schedule a final audit), **No** (no final audit is required), or **Determined By Business Rule**. When a final audit is selected or required, the application determines which final audit schedule to use.

When the policy is issued, PolicyCenter adds *audit scheduled items* to the policy. You can see these audits by clicking the **Audit Schedule** link of the policy file. Initially these are not audit policy transactions, rather they are a list of all the audit policy transactions anticipated but not yet initialized for the policy period. They are listed

according to their start and end dates and their status is **Scheduled**. When a policy is canceled or reinstated, PolicyCenter revises the number of audits scheduled according to the coverage dates.

Each of the scheduled items includes a process start date, an audit method and a due date. Users with the proper permissions can edit these fields before the premium audit policy transaction is initialized. For example, you can change a final audit with an audit method of voluntary to physical. You can also waive a premium audit. However, the final audit may not be waived on a premium reporting policy because it is the mechanism to return the initial deposit.

A regularly scheduled batch process called **Audit Task** starts the audit policy transactions on their process start date. The audit status changes from scheduled to in progress. The audit becomes a draft policy transaction and the scheduled item becomes a link to the audit wizard. You can begin entering the audit information.

After receiving the audit details, you can enter them into the **Audit Summary** and **Audit Details** screens. Determine the premiums by selecting the **Calculate Premiums** button. Finalize the calculations by selecting the **Submit** button. At that time, the audit becomes uneditable, the status becomes **Completed**, and the audit schedule displays the resulting premiums.

You can change completed audits. Premium reports can be manually reversed and rebilled. Final audits are automatically reversed and rescheduled by policy changes completed after the final audit. Final audits can also be manually revised. A revised audit displays current and previous premium values.

Premium Report Audit Schedules

Premium report audit schedules include settings that allow for a wide variety of audit schedules. Just like final audit, there are schedule settings for process start date, due dates, and audit methods. In addition, premium reporting audit schedules contain settings for the following:

- The frequency of reports
- Whether the report dates match calendar dates or policy dates
- The minimum period length
- Whether to include or exclude the last period

The premium report items scheduled for a particular policy may be amended by changes to the policy such as a cancellation or change to the effective or expiration date.

Premium Report Trend Analysis

In a policy with premium reports, *reporting trend analysis* is a way to track how far apart the reported premiums are from the estimated premiums. Reporting trend analysis tracks the total estimated premium, total reported premium, the ratio between the premiums, and number of days reported. PolicyCenter displays these values on the policy **Summary** page and premium report **Premiums** page. These values are also stored in the database when the audit is submitted or after a premium report reversal. The database is also updated after a policy change, cancellation or reinstatement. You may use out-of-range ratios to indicate that the underwriter needs to review the policy for accuracy.

See also

- “Entering Premium Report Data” on page 158 to view the trend analysis information in the application
- “Rules Sets” bullet item in “Configuring Premium Audit Overview” on page 595 in the *Configuration Guide* for information on the code that determines the ratio and creates the activity

Adding a Premium Report

PolicyCenter allows you to add a premium report to a policy for any unreported portion of a policy period. For example, you may add a premium report to a policy in the following cases:

- The policy originally had 11 monthly reports plus a final audit. The underwriter wants to add a 12th report rather than waiting for the final audit to collect the premium for the last month.
- A user submits a policy change on a policy with one or more completed reports. These reports need to be reversed and added back.

See also

- “Adding a New Premium Report” on page 158

Audit Roles

Many regular users of PolicyCenter are underwriting personnel or producers. Although premium audits interest these users, the actual initiation and processing of premium audits is typically done by a separate premium audit department. The premium audit department might include in-office *premium audit examiners*, and outside *premium auditors* who travel to the policyholder’s location.

The default application contains the following audit roles:

- Premium Auditor
- Premium Audit Examiner
- Premium Audit Supervisor

Premium Auditor

PolicyCenter assigns an audit with a method of *physical* to a premium auditor. The premium auditor is a person who travels to the policyholder’s location to conduct the audit. The premium auditor returns that information to the carrier’s office. The premium auditor role can vary by carrier. For example, the auditor may be allowed to edit the audit summary and audit details. However, the auditor may not be allowed to complete the audit because a premium audit examiner needs to check it for accuracy.

Note: In the default configuration when the audit method is physical, PolicyCenter assigns the audit to a user with the premium auditor role.

Premium Audit Examiner

The premium audit examiner is located at the office of the carrier. The premium audit examiner enters the audit details for premium reports and final audits. The premium audit examiner checks physical audits for accuracy, calculates the premiums, and completes the audit.

If the final audit method is:

- **Physical** – The premium audit examiner receives the physical audit from the premium auditor and checks it for completeness and accuracy. The examiner also checks it for adherence to manual rules and guidelines of the insurance company. For example, does it include all the locations and entities? Does it split exposures for the anniversary rating date (ARD)?
- **Voluntary** – The premium audit examiner receives a completed report form from the policyholder. The examiner reviews the form prior to billing to ensure that the policyholder completed it properly. If necessary, the premium audit examiner calls the policyholder to clarify or amend the totals received.
- **By phone** – The premium audit examiner calls the policyholder to obtain audit exposures and ask questions appropriate to the audit or renewal.

Premium Audit Supervisor

The premium audit supervisor is a supervisor in the premium audit department. Premium audit supervisors have all the audit permissions. In addition the supervisor has access to team screens where outstanding audits and audit activities can be tracked.

Premium Audit and Other Policy Transactions

The following table describes the interaction between final audit and premium report and other policy transactions.

Policy transaction	Description
Audit	Premium Report Policy Transactions If you have an open report and final audit is already billed, then the report cannot be billed because the final audit already finalized the premium for this period. This situation occurs if the auditor does the final audit early and completes it before receiving the last report.
Submission	Final Audit and Premium Report Policy Transactions According to the configuration settings, PolicyCenter schedules final audit and appropriate premium reports when you issue the policy.
Issuance	Final Audit and Premium Report Policy Transactions If changes are made to the policy term, then PolicyCenter schedules final audit and appropriate premium reports when issuing the policy.
Renewal	Final Audit and Premium Report Policy Transactions PolicyCenter schedules a final audit and appropriate premium reports when renewing the policy.
Policy Change	Final Audit Policy Transaction Policy changes impact final audits as follows: <ul style="list-style-type: none">• Scheduled audits – If the policy change amends the policy term, PolicyCenter replaces the scheduled audit with an audit for the new policy term.• In Progress audits – PolicyCenter preempts and updates the In Progress audit. The updated audit includes the policy changes, such as classification or policy term changes.• Completed audits – PolicyCenter reverses the Completed audit and schedules an audit for the full policy term or cancellation period. If the Completed audit has an In Progress revision, PolicyCenter withdraws the revision. Premium Report Policy Transaction <ul style="list-style-type: none">• Scheduled premium reports – if the policy change amends the policy term, PolicyCenter replaces the audit schedule with the appropriate premium report periods.• In Progress premium reports – If there is an open premium report, a policy change preempts the report and adds in the changes.• Completed premium reports – A policy change does not impact a Completed premium report.

Policy transaction	Description
Cancellation	<p>Audits are affected when the policy reaches a Canceled status. Cancellations may be completed with a scheduled future effective date or canceled immediately. Policies canceled immediately are given a Canceled status. Policies with Scheduled cancellations reach a Canceled status on the cancellation date.</p>
	Final Audit Policy Transaction
	<p>When the policy changes from a Canceling to a Canceled status, audits are impacted as follows:</p>
	<ul style="list-style-type: none"> • If the policy is canceled flat, no audit is required. PolicyCenter removes any scheduled final audit or withdraws any open final audit. • If the cancellation is midterm: <ul style="list-style-type: none"> • Scheduled audits – PolicyCenter replaces the full term audit in the schedule with an audit for the cancellation period, including assigning the configured audit schedule. The cancellation calculates the cancellation amount and sends this amount to the billing system. The cancellation also sends a message to hold these funds until final audit completes. • In Progress audits – PolicyCenter preempts the In Progress audit. PolicyCenter amends the dates and displays only classifications that apply to the cancellation term. • Completed audits – PolicyCenter reverses the Completed audit and schedules an audit for the cancellation period. If the Completed audit has an In Progress revision, then PolicyCenter withdraws the revision.
	Premium Report Policy Transaction
	<p>When the policy changes from a Canceling status to a Canceled status, premium reports are impacted as follows:</p>
	<ul style="list-style-type: none"> • Scheduled audits past the cancellation date are removed. • Completed reports remain completed regardless of the period they cover. • In Progress reports <ul style="list-style-type: none"> • Any reports Scheduled or In Progress with dates prior to a completed report remain regardless of the cancellation date. • An In Progress period that includes the cancellation date is Withdrawn. • In Progress periods later than the cancellation date are Withdrawn. • In Progress reports covering the period prior to the report that includes the cancellation date are Withdrawn if both of the following are true. 1) If the premium report schedule excludes the last month. 2) The number of days from the end of the prior period to the cancellation date is less than the minimum audit period length.
Reinstatement	Final Audit Policy Transaction
	<p>Reinstatement policy transactions impact audits as follows:</p>
	<ul style="list-style-type: none"> • Scheduled audits – PolicyCenter replaces the cancellation period audit with an audit for the full policy period. • In progress audits – PolicyCenter withdraws the in progress cancellation audit and schedules a new audit for the full policy term. • Completed audits – PolicyCenter reverses a completed audit and schedules an audit for the full policy term.
	Premium Report Policy Transaction
	<p>Reports are reinstated as follows:</p>
	<ul style="list-style-type: none"> • Periods that were Withdrawn, Waived, or removed are added again as new Scheduled items • The Waived and Withdrawn scheduled items remain in the list as Waived and Withdrawn. • Completed periods that were reversed are also Scheduled as new scheduled items.
Rewrite	Final Audit Policy Transaction
	<p>When the rewrite finishes, PolicyCenter schedules a final audit for the new period.</p>
	Premium Report Policy Transaction
	<p>According to the configuration settings, PolicyCenter schedules premium reports when you issue the rewritten policy.</p>

Note: Final audits cannot be out of sequence with other policy transactions. However, final audits can be preempted.

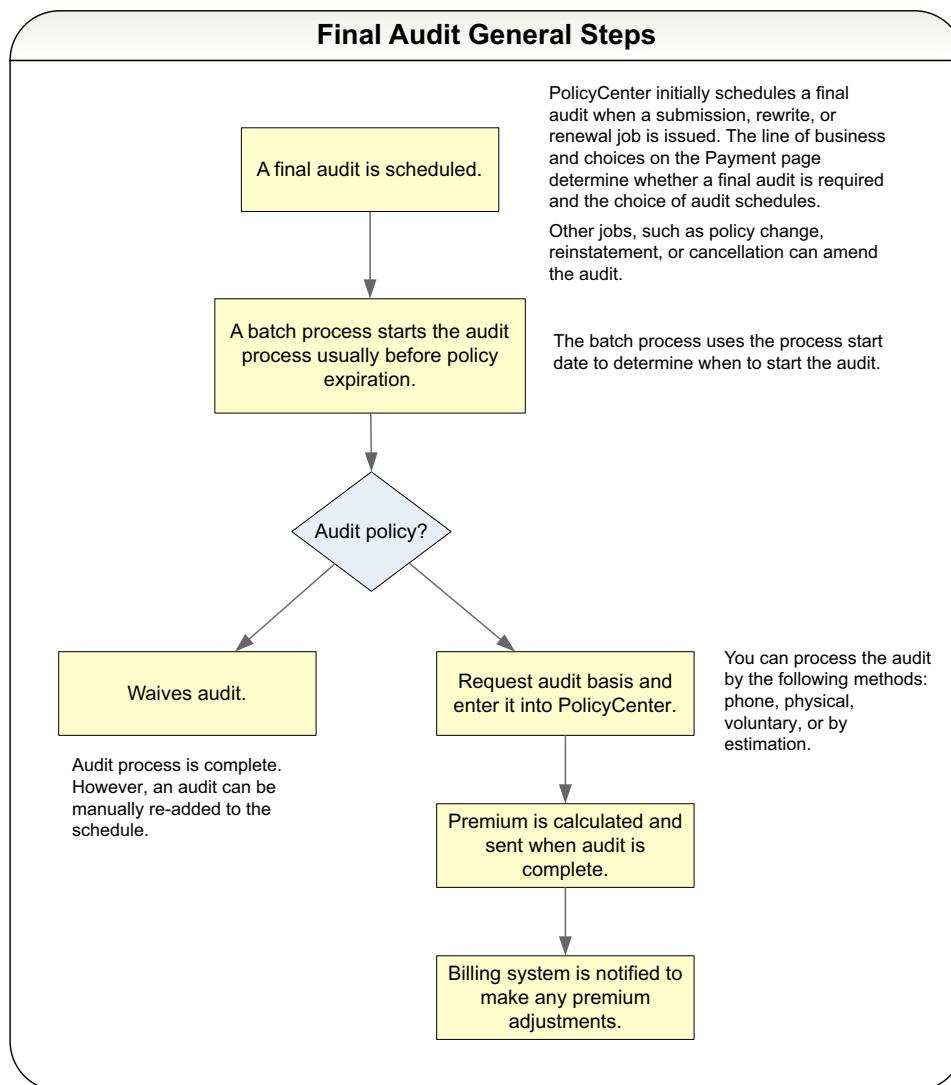
Premium Audit General Steps

The sections describes the general steps for premium audit.

- Final Audit General Steps
- Premium Reports General Steps

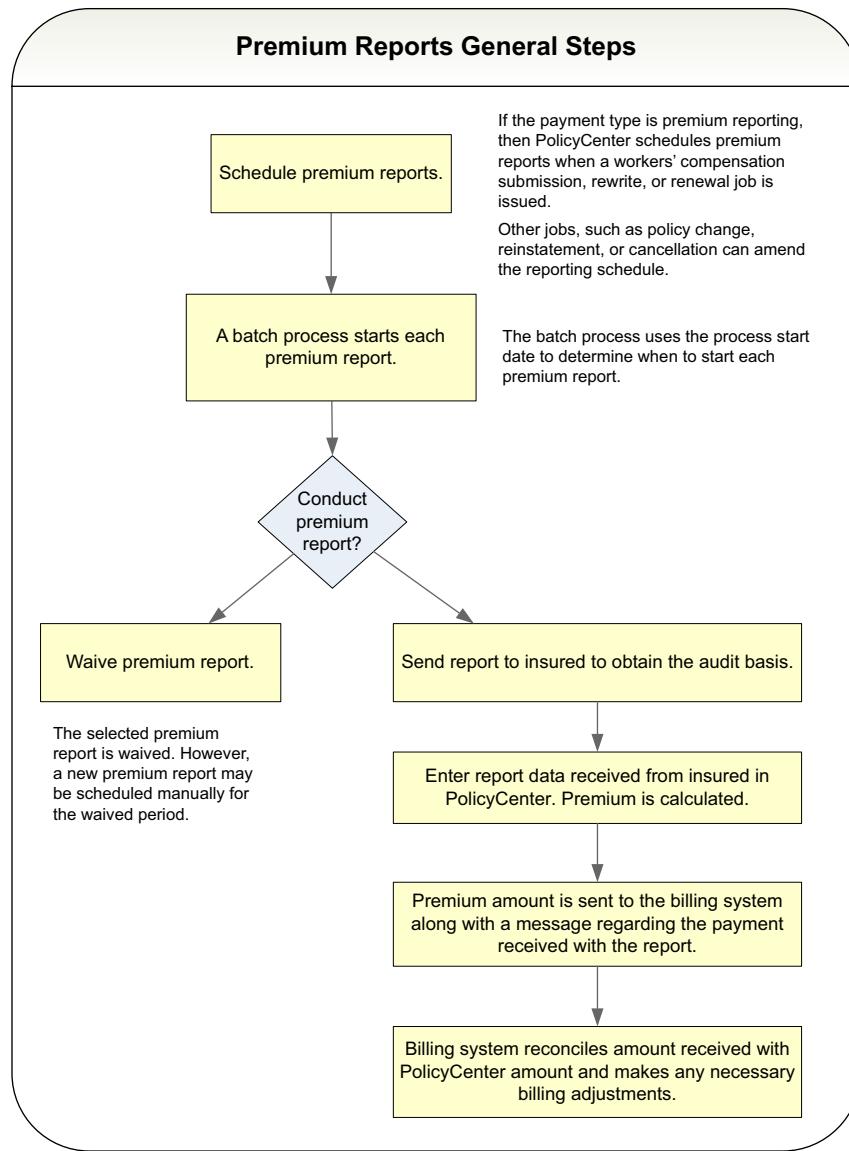
Final Audit General Steps

The following diagram shows the main steps that are necessary to conduct a final audit. These steps are for the default configuration of PolicyCenter. Your business requirements can alter the process.



Premium Reports General Steps

The following diagram shows the main steps that are necessary to conduct premium reports. These steps are for the default configuration of PolicyCenter. Your business requirements can alter the process.



Working with Final Audits

You can schedule a final audit upon issuance of a workers' compensation or general liability submission. Or you can manually schedule an audit later in the policy term. Close to policy expiration, PolicyCenter creates the final audit policy transaction according to the process start date. You can configure when the final audit policy transaction starts by configuring final audit schedule patterns. For more information on configuring audit, see “Audit Schedules” on page 596 in the *Configuration Guide*.

If you have the appropriate audit permissions, you can edit, calculate, complete, or revise a final audit. By default, a producer or underwriter can view some details of the final audit but cannot edit it.

This topic provides step-by-step instructions for the following:

- “Viewing a Final Audit” on page 153
- “Viewing Audit Activities” on page 153
- “Editing or Waiving a Final Audit” on page 154
- “Adding a New Audit” on page 154
- “Starting a Final Audit” on page 154
- “Entering Audit Data and Completing a Final Audit” on page 155
- “Revising a Final Audit” on page 155

See also

- “Policy Tools Menu” on page 307

Viewing a Final Audit

You can view basic details about a final audit, such as the period start and end dates, the method of the audit, and its status.

To view a final audit

1. Navigate to a policy.
2. Select **Audit** in the **Tools** menu. The **Audit** screen displays a summary of audits and audit scheduled items (future audits).

Note: Without sufficient permissions, you can only view summary information.
3. Select the pull-down menu to filter the list of audits. The menu displays the following options:
 - **Scheduled/in-progress** – Default. Display open audit policy transactions or audits that are scheduled for the future.
 - **In progress** – Display only open policy transactions.
 - **Closed within last 12 months** – Display any item that has a closed date within the last 12 months.
 - **Due date within last 12 months** – Display any item with a due date within the last 12 months or due in the future.
 - **End date within last 12 months** – Display any item that has an end date in the last 12 months or a future end date.
 - **All** – Full history of audit policy transactions excluding deleted policy transactions.

Viewing Audit Activities

The audit batch process changes the status of an audit to open and assigns the audit. For each audit, the batch process creates an activity with the subject **A new audit has been assigned**. You can find these activities by selecting **Activities** from the **Search** tab.

To find audit activities that are assigned to you

1. Navigate to the **Desktop** tab and click **Activities** in the left sidebar. Look for activities with the subject **A new audit has been assigned**.
2. Click the **Subject** of an audit activity. The audit appears at the top of the screen, and the Activity appears at the bottom.

You can complete the audit and close the activity. For instructions on how to complete the audit, see “Entering Audit Data and Completing a Final Audit” on page 155.

Editing or Waiving a Final Audit

If a final audit is scheduled (but has not started yet) you can edit details of how it is scheduled, or waive it. Follow these steps when you need to waive an audit or change any of the dates or methods for an upcoming final audit.

Note: You must have sufficient permissions to edit or waive an audit. In the base configuration, the **Edit** and **Waive** buttons appear for users with the **Audit Supervisor**, **Audit Examiner** or **Premium Auditor** role.

To edit or waive a final audit

1. Perform the same steps as in “Viewing a Final Audit” on page 153. Under the **Actions** column, you can select either **Edit** or **Waive**.

If you select **Waive**, you have decided to bypass the audit procedure. PolicyCenter will not perform an audit on that period. If you later decide that you need the audit, you can add it back in. See “Adding a New Audit” on page 154.

Note: When an audit is in progress, the **Edit** and **Waive** buttons are not available. Changes or waives must be done within the audit. See “Entering Audit Data and Completing a Final Audit” on page 155.

2. If you select **Edit**, you can edit the **Process Start** date, **Due Date**, or planned **Method**.

Adding a New Audit

In certain cases, you can add a new audit or premium report. The section “Adding a Final Audit” on page 146 describes when you can add a new audit. The section “Adding a Premium Report” on page 147 describes when you can add a new premium report.

To add a new audit

1. Navigate to a policy that has an audit or premium reports.
2. Click **Audit Schedule** in the left sidebar.
3. Click **New Audit**.
4. Select appropriate values and click **Update**.

The screen for final audit allows you to enter **Process Start Date**, **Due Date**, and **Audit Method**. **Audit Period Start Date** and **Audit Period End Date** are set automatically to the start and end of the period.

The screen for premium report allows you to enter **Audit Period Start Date**, **Audit Period End Date**, **Process Start Date**, **Due Date**, and **Audit Method**.

Starting a Final Audit

The **AuditTask** batch process starts final audits. In the base configuration, the audit batch process runs at a predefined interval.

For more information on batch processes and how to configure them, see “Batch Processes and Work Queues” on page 91 in the *System Administration Guide*.

Entering Audit Data and Completing a Final Audit

If a final audit has a status of **In Progress**, you can enter the audit data. The following steps describe how to edit and complete an audit.

Note: You must have sufficient permissions to edit and complete an audit. In the base configuration, the **Premium Auditor** role has permissions to edit an audit (`editaudit` permission code) and calculate premiums (`quote` permission code). The **Audit Examiner** role has permissions to edit, quote, and complete an audit (`completeaudit` permission code).

To enter audit data and complete a final audit

1. Log in to PolicyCenter.
2. Perform the steps in “Viewing a Final Audit” on page 153 or “Viewing Audit Activities” on page 153. Then click the link of the **Final Audit** you wish to complete.
3. Enter information in the **Summary** screen.

The **Audit Summary** screen allows you to enter the following:

Field	Description
Due Date	Modify the due date, if necessary.
Received Date	Enter the date that the audit information was received.
Method (actual)	Change the audit method to the type of audit that actually occurred. In most cases this is the same value as Method (planned) , however, sometimes it is not. If a physical audit was attempted but did not succeed, then this audit policy transaction may need to be completed with Estimated values.
Audit Fee	Enter an audit fee when a vendor conducts the audit.
Instructions	Enter instructions for the premium auditor or premium audit examiner.

4. Click **Next**.
5. In the audit **Details** screen, all the exposure class codes for the period appear by jurisdiction. Enter the actual audited payroll amounts for each location and class code. Click **Save Draft**.
Because of your permissions, the screen displays **Calculate Premiums** next to the **Save Draft** button.
6. Click **Calculate Premiums**. The rating engine calculates the amounts and displays the results on the **Summary** tab of the audit **Premiums** screen.
The **Summary** tab displays the audit premium and the difference between the audited costs and the costs on the policy. Use the **Comments** field to add an explanation for the difference.
7. To see the calculation of the total premium for the audit term, select the **Premium Details** tab.
The **Premium Details** tab shows the calculation for each audit cost and compares it to the policy contract values. You can see the difference between these costs in the **Change** column.
8. If you need to change the class code exposures, click **Edit Audit**.
9. When the amounts are accurate, click **Submit**. The audit status is now **Completed**.

Revising a Final Audit

If a final audit has a status of **Completed**, you can revise the audit through a policy transaction called a revised final audit.

You must have sufficient permissions to start an audit revision. In the base configuration, the **Audit Examiner** and **Premium Auditor** roles have these permissions.

IMPORTANT Starting a revision policy transaction does not submit a change to the billing system. It is only upon completion of the audit revision that the billing system is notified of the change in final audited premiums.

To revise a final audit

1. Perform the steps in “Viewing a Final Audit” on page 153, and click **Revise** under the **Actions** column. PolicyCenter starts the audit policy transaction. This action also changes the original audit to a status of **Revised**.

2. Enter revised information as you would in the “Entering Audit Data and Completing a Final Audit” on page 155.

Note: The revised final audit policy transaction displays **Close Options → Withdraw Transaction** instead of a **Close Options → Waive**. If you select **Withdraw Transaction**, the policy transaction goes to a **Withdrawn** status. When a revision is withdrawn, PolicyCenter changes the status of the revision policy transaction to **Withdrawn** and returns the status of the original audit to **Completed**.

3. Click **Calculate Premiums**.

The **Audit Premiums → Summary** tab displays the difference between the original audit and the revised audit. To see a breakdown of the amounts, select the **Premium Details** tab.

4. Click **Submit** to complete the audit.

Working With Premium Reports

You can choose premium reports on the **Payments** screen in the workers’ compensation line of business. This topic describes how to work with premium reports in PolicyCenter.

- “Viewing a Premium Report” on page 156
- “Viewing Premium Report Activities” on page 157
- “Selecting Premium Reports” on page 157
- “Starting a Premium Report” on page 157
- “Creating Activities for Overdue Premium Reports” on page 157
- “Entering Premium Report Data” on page 158
- “Editing or Waiving a Premium Report” on page 158
- “Adding a New Premium Report” on page 158

See also

- “Policy Tools Menu” on page 307

Viewing a Premium Report

You can view basic details about a premium report, such as the period start and end dates, the method of the audit, and status. Viewing a premium report is the same as viewing a final audit. See “Viewing a Final Audit” on page 153.

Viewing Premium Report Activities

The **AuditTask** batch process changes the status of a premium report to open and assigns it. For each premium report, the batch process creates an activity with the subject **A new audit has been assigned**.

The **Overdue Premium Report** batch process creates an activity for overdue premium reports. PolicyCenter displays the activity on the policy **Summary** page under **Current Activities** or on the underwriter's **Desktop** under **My Activities**.

The display of activities is the same as for final audit. See “Viewing Audit Activities” on page 153.

See also

- “Starting a Premium Report” on page 157 for more information about the **AuditTask** batch process.
- “Creating Activities for Overdue Premium Reports” on page 157 for more information about the **Overdue Premium Report** batch process.

Selecting Premium Reports

You can select premium reports on the **Payment** screen of a workers' compensation submission, rewrite, or renewal policy transaction.

To select premium reports

1. Start a submission, rewrite, or renewal policy transaction and navigate to the **Payment** screen.
2. In **Payment Method**, select **Reporting Plan**.
3. Under **Premium Report Plans**, select one of the plans. In this example, select the first plan, **Monthly Reports by calendar months, excl. last month**.

In the base application, PolicyCenter displays the following **Premium Report Plans**:

- **Monthly reports by calendar month, excl. last month**
- **Monthly reports by policy month, excl. last month**
- **Quarterly reports by calendar quarters**
- **Quarterly reports by calendar quarters, excl. last quarter**
- **Quarterly reports by policy quarters, excl. last quarter**

After you choose a report plan, PolicyCenter displays fields for the deposit percentage and amount. Deposit percentage is configured in the audit schedule. You can override the percentage in the **Deposit override %** field.

All report plans in the base application require final audit. The final audit notifies the billing system to release the deposit.

4. After you issue the policy, you can view the audit schedule. In the **Tools** sidebar, click **Audit Schedule**.

Starting a Premium Report

The **AuditTask** batch process starts premium reports. In the base configuration, PolicyCenter launches the audit batch process at a predefined interval. The batch process activates premium reports that have passed their process start date. These reports have an **In progress** status in the **Audit Schedule**.

For more information on batch processes and how to configure them, see “Batch Processes and Work Queues” on page 91 in the *System Administration Guide*.

Creating Activities for Overdue Premium Reports

Premium reports have a process start date, which is the date when the audit task batch process can start processing the report. If the user does not make the payment by due date then the premium report goes into delin-

quency. The **Overdue Premium Report** batch process finds overdue premium reports. For each overdue premium report, the batch process creates a **Premium report overdue** activity for the underwriter.

Entering Premium Report Data

The insured completes the premium report and returns it to the carrier. The user enters the report data into PolicyCenter.

To enter premium report data

1. Navigate to a policy with premium reports.
2. Click **Audit Schedule** in the left sidebar.
If the batch process has started a premium report, **Premium Report** is a link in the **Type** column.
3. Click a **Premium Report** link.
4. In the **Summary** screen, enter the **Received Date** and **Payment Received**. Then click **Next**.
5. In the **Details** screen, enter the payroll amounts reported by the customer.
6. Click **Calculate Premiums**.

The **Premiums** screen displays the calculated cost along with the payment received from the insured.

The **Reporting Trend Analysis** displays the ratio between the total estimated premium and total reported premium. The Reporting Trend Analysis also appears on the policy **Summary** screen. For more information about reporting trend analysis, see “Premium Report Trend Analysis” on page 147.

Editing or Waiving a Premium Report

You can choose to edit or waive a premium report. The steps are the same as for final audit. See “Editing or Waiving a Final Audit” on page 154.

Adding a New Premium Report

In certain cases, you can add a new premium report. The section “Adding a Premium Report” on page 147 describes when you can add a new premium report. The steps are similar to the steps for final audit. See “Adding a New Audit” on page 154.

Side-by-side Quoting

With side-by-side quoting, you can view multiple versions of a policy transaction on one screen. You can modify the coverages and terms of each version in the side-by-side screen, and see the side-by-side comparison of the costs and benefits of each version. You can use side-by-side quoting with quick quote. In the default configuration, the personal auto line of business provides side-by-side quoting. You can configure side-by-side quoting for other lines of business.

This topic includes:

- “Side-by-side Quoting Overview” on page 159
- “Side-by-side Quoting in the User Interface” on page 162
- “Working with Side-by-side Quoting” on page 165

See also

- “Configuring Side-by-side Quoting” on page 601 in the *Configuration Guide*

Side-by-side Quoting Overview

This topic provides a high-level description of side-by-side quoting.

- “Side-by-side Quoting Compared with Multi-Version Quoting” on page 159
- “Side-by-side Availability Overview” on page 160
- “Base Data Overview” on page 160
- “Side-by-side Data Overview” on page 161
- “Side-by-side Quoting Process Flow” on page 162

Side-by-side Quoting Compared with Multi-Version Quoting

PolicyCenter provides two ways to generate alternate versions of a policy quote: side-by-side quoting and multi-version quoting. The two options are mutually exclusive.

The following table compares and contrasts some of the features of side-by-side quoting and multi-version quoting.

Side-by-Side Quoting	Multi-Version Quoting
Side-by-Side Quoting screen allows you to view and modify multiple quotes in one place.	Select version from drop-down menu under Actions. No side-by-side comparison view.
Copies changes to base data to other side-by-side versions.	All policy data can differ. Provides more flexibility than side-by-side quoting.
When you switch to side-by-side quoting, PolicyCenter quotes each version.	When you create a new version, PolicyCenter does not quote either version of the policy.
Use the Quote All button to quote all versions.	Quotes each version of the policy independently.
Available in the personal auto line of business in the default configuration.	Available in all lines of business except personal auto in the default configuration.

See also

- “Multi-version Quoting” on page 169

Side-by-side Availability Overview

PolicyCenter supports side-by-side quoting for the following policy transaction types:

- Submission
- Policy change
- Renewal

The policy transaction must have a status of:

- New
- Draft
- Quoting
- Quoted

You cannot enter side-by-side quoting in a policy change or renewal that has out-of-sequence conflicts or unhandled preemptions. PolicyCenter displays a warning message after you click the **Side-by-Side** button to enter side-by-side mode.

Base Data Overview

Each line of business which implements side-by-side quoting defines *base data*, which data that is common across all versions. Making a change to the base data in one side-by-side version triggers a change to the base data in all side-by-side versions. You can change the base data in any policy transaction wizard step, such as on the **Policy Info** screen in a submission. However, you cannot edit the base data in side-by-side view.

IMPORTANT To conform to Guidewire configuration requirements, base data entities or fields on the **Side-by-Side Quoting** screen must not be editable in more than one place on a given screen. For example, placing an editable widget for a base data field in the columns replicated for each version is a violation of this requirement. This requirement applies to fields that are implicitly base data, such as contact or location information that can be synchronized.

Typical types of base data on the policy period include:

- Account associated with this policy
- Policy
- Effective dates

- Answers to pre-qualification questions
- Address and contact information

In the personal auto line of business, the base data also includes line-level modifiers.

Side-by-side Data Overview

A change to side-by-side data affects only a single version. PolicyCenter does not propagate the change to other side-by-side versions.

Typical Side-by-side Data

Typically side-by-side data includes:

- Coverages, exclusions, and conditions
- Forms
- Underwriting issues and approvals
- Costs and transactions
- Other financial data
- Numbering of coverables
- Reinsurance-related information
- Grandfathering-related dates
- Internal status information
- Denormalized data
- Archiving- and purging-related information

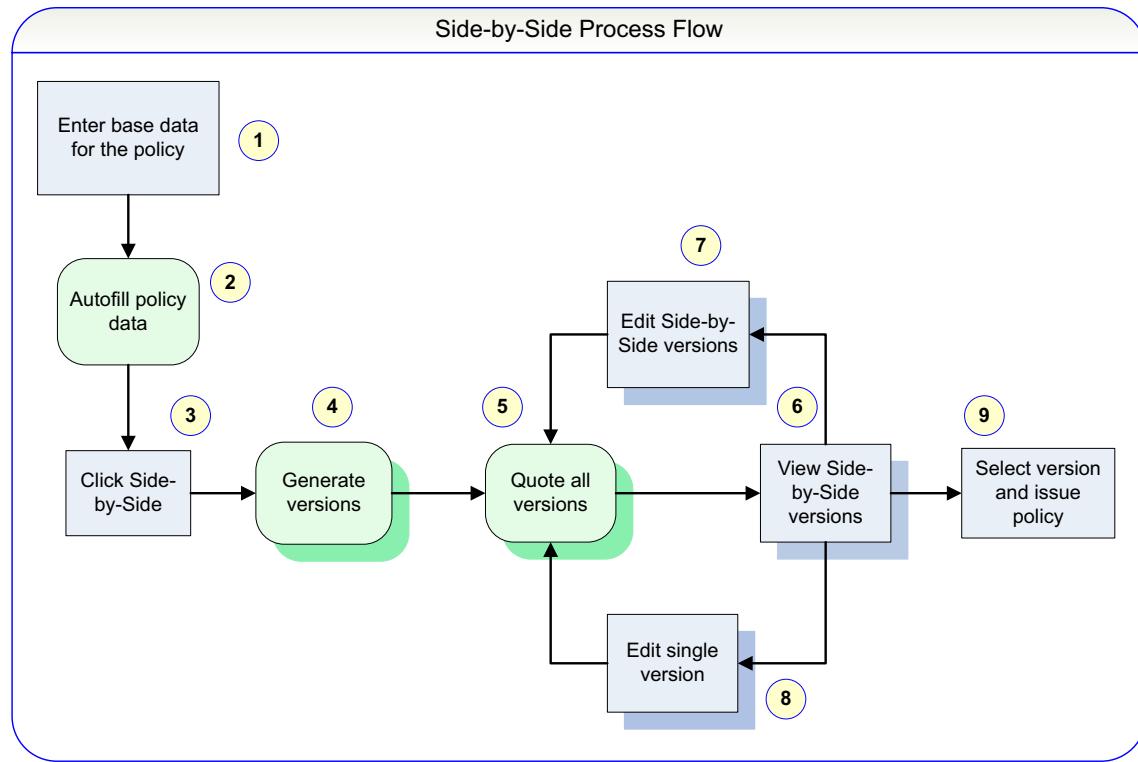
Personal Auto Side-by-side Data

In the personal auto line of business, the side-by-side data includes:

- The selected offering code
- Line-level coverages, such as liability coverage
- Vehicles, along with their coverages
- Vehicle drivers
- Personal auto vehicle additional interest
- Quick quote numbering

Side-by-side Quoting Process Flow

The following illustration shows the process flow of side-by-side quoting.



1. In PolicyCenter, the user enters base data for the policy such as the primary named insured and contacts on the policy.
2. PolicyCenter automatically fills in territory codes and similar data.
3. The user clicks **Side-by-Side** to enter side-by-side mode.
4. PolicyCenter generates multiple versions of the policy by using business logic configured for the line of business. The number of versions is also configurable.
5. PolicyCenter generates quotes for all versions.
6. The user views a screen showing side-by-side versions of the policy.
7. The user makes modification to the side-by-side data, such as making changes to coverages. Then to compare the costs of the different versions, the user quotes the policy (step 5).
8. Alternately, the user can edit a single version and change the base data or side-by-side data. PolicyCenter copies changes to base data to the other side-by-side versions. Then to compare the costs of the different versions, the user quotes the policy (step 5).
9. The user selects a version and issues the policy.

Side-by-side Quoting in the User Interface

This topic describes the screens and menu items in the user interface for side-by-side quoting and contains the following:

- “Tools Menu Items for Side-by-side Quoting” on page 163
- “Side-by-side Quoting Screen” on page 163

- “Policy Versions Screen” on page 165

Note: The descriptions use the personal auto line of business.

Tools Menu Items for Side-by-side Quoting

If you are in a policy transaction enabled for side-by-side quoting, two additional menu items appear in the **Tools** menu in the left sidebar:

- **Side-by-Side Quoting** – Displays the **Side-by-Side Quoting** screen.
- **Policy Versions** – Displays the **Policy Versions** screen which contains brief information about each side-by-side version. This screen also appears for multi-version quoting policy transactions.

Side-by-side Quoting Screen

On the **Side-by-Side Quoting** screen, you can view and modify multiple versions of a policy and compare quotes for each version. For each version, this screen displays side-by-side data fields which you can modify.

Side-by-side Versions Locked

In some circumstances, certain users cannot modify any versions in a side-by-side policy transaction. If there are one or more locked versions, and you do not have the **Edit Lock Override** permission, you cannot make modifications until there are no locked versions. For Gosu code, you can use the `UWLockedAndNoOverride` property on a `PolicyPeriod` to determine if this lock applies to a user. Locking across versions prevents inconsistencies in sharing base data across versions in side-by-side mode. The `UWLockedAndNoOverride` property is defined in `gw.policy.PolicyPeriodBaseEnhancement`.

For example, if you select **Request Approval** for one or more versions on the **Risk Analysis** tab, those versions are locked awaiting underwriter approval. Consequently, all versions in the side-by-side policy transaction are also locked. If you do not have the **Edit Lock Override** permission, you cannot make modifications until underwriting approves all versions. If PolicyCenter did not lock all versions, you could modify base data in an unlocked version, but PolicyCenter could not copy those changes to the locked versions. If you have the **Edit Lock Override** permission, you have permission to modify all versions, including locked versions. Therefore, if you make a change, base data copy copies your change to the other versions.

Buttons at Top of Screen

Some of the buttons at the top of this screen are:

- **Add Side-by-Side Version** adds a new side-by-side period up to a configurable limit. If you click **Select** next to a version to jump to the single version wizard for that period, this button appears in the **Versions** drop-down menu. On the **Side-by-Side Quoting** screen, the selected policy period is the basis of the new version. In the single version wizard, the current policy period is the basis of the new version.
See also “[Changing the Maximum Number of Versions in Side-by-side Quoting](#)” on page 601 in the *Configuration Guide*.
- **Quote All** generates quotes for all versions.
- **Save All** saves all versions.

Quote Comparison

Under the **Personal Auto Quote Comparison** label, each version has the following:

- **Policy Period Branch Name** – A text box for editing the branch name. A star appears next to the text box of the selected version.
- **Selected** – This word appears in the **Policy Period Branch Name** row next to the selected version.
- **Offering Selection** – Has a drop-down menu to select an offering.

- **Reset** – Applies the selected offering to the policy period, setting all coverages to default values in the product model.

When you click **Reset**, PolicyCenter synchronizes the product model with the currently selected offering. Simply changing the offering selection does not synchronize the product model for the current version.

- Validation errors and underwriting issues – Appear in the final row under the **Personal Auto Quote Comparison** label. You cannot **Quote All** if a version has validation errors or underwriting issues that block quote. In many cases, a wizard step is selected based on the condition needing resolution.
- **Select** – Select this version. When you click **Select**, PolicyCenter takes you to the **Policy Review** screen for that version. PolicyCenter marks the selected version with an asterisk in the drop-down list underneath the **Actions** menu. You can navigate backwards in the policy transaction wizard to make changes to other screens, including changes to base data and side-by-side data. In the **Side-by-Side Quoting** screen, the word **Selected** appears next to the **Policy Period Branch Name** for the selected version.

Clicking **Select** marks the selected policy period for the policy transaction. Reporting and other PolicyCenter processes may access this status. Copies changes to base data are copied to the other versions. For example, in personal auto, you can go back to the **Policy Info** screen and change the **Effective Date**. This change to base data is copied to all side-by-side versions. Select **Versions** → **View Side-by-Side Versions** to return to the **Side-by-Side Quoting** screen.

- **Withdraw** – Withdraw that version and only that version.
- **Resolve** – Forwards you to a step in the policy transaction wizard for the version based on the type of validation errors or underwriting issues. You can edit the policy and make changes in that step. If errors are present, PolicyCenter displays a worksheet with error or warning messages and, in many cases, links to the step in the wizard. This behavior is similar to the display of errors and warnings when attempting to quote.

Coverages

The bottom of the **Side-By-Side Quoting** screen displays coverages for each version. You can make coverage selections for each version.

Entering the Side-by-side Quote Screen for the First Time

You enter side-by-side for the first time by selecting **Versions** → **Start Side-by-Side**. If you select **Start Side-by-Side** in the default configuration, PolicyCenter creates two additional periods for a total of three side-by-side periods. If the initial period has an offering, the side-by-side periods are the same as the initial period. If there is no offering on the original period, PolicyCenter applies the basic, standard, and premium offerings to each side-by-side version, respectively.

In policy periods that have the **Standard Program** offering, PolicyCenter removes collision and comprehensive coverages on vehicles over 10 years old.

If an underwriter locked the policy, then **Policy Transaction Under UW review** appears in the info bar and the agent cannot view quotes for the versions. An underwriter might lock the policy so that the agent cannot make changes while the policy is under review.

Resolving Out-of-sequence Conflicts and Unhandled Preemptions

If there are out-of-sequence conflicts or unhandled preemptions, the **Side-by-Side Quoting** screen displays a warning message and does not create multiple versions. If the policy transaction is already in side-by-side mode, then PolicyCenter prevents you from editing the side-by-side data.

If there are unhandled preemptions, you can apply all the preemptions or withdraw the changes. If you apply preemptions in a side-by-side policy transaction, PolicyCenter applies preemptions against the current policy period. PolicyCenter then copies base data from the current policy period to the other side-by-side periods. If you choose to withdraw the changes, the policy transaction is marked for withdrawal.

In the HandlePreemptionPopup PCF file, the applyChanges method handles preemption in side-by-side policy periods. You can view the HandlePreemptionPopup in Studio by navigating to configuration → config → Page Configuration → pcf → job.

If there are out-of-sequence conflicts, you resolve these conflicts in the single wizard view. When you resolve out-of-sequence conflicts for a particular policy period, PolicyCenter copies base data product model changes to the other side-by-side periods. In some cases the copy does not fix out-of-sequence conflicts for other side-by-side periods. If so, PolicyCenter forwards you to the out-of-sequence conflict resolution page of a period that still has conflicts.

The handleConflict method in the OOSConflictPanelSet PCF file handles out-of-sequence conflicts in a side-by-side policy.

Policy Versions Screen

The **Policy Versions** screen displays brief information about each version. In side-by-side quoting policy transactions, a link to this screen is available under **Tools**.

Buttons at Top of Screen

The following buttons appear at the top of the screen:

- **Select** – To make a version the selected version, select a version in the **Status** row and click this button.
- **Rename** – To rename a version, select a version in the **Status** row and click this button.
- **Withdraw** – To withdraw one or more versions, select versions in the **Status** row and click this button.
If all but one period is withdrawn, the policy transaction is taken out of side-by-side mode, and the **Side-by-Side Quoting** screen is no longer available. The **Versions** → **Start Side-by-Side** menu item is enabled.
If you select all versions and click **Withdraw**, PolicyCenter displays a message in the **Validation Results** and does not remove any versions.
- **Diff** – To see the differences between two versions, select two versions in the **Status** row and click this button.

Each Version

For each version in a side-by-side quote, the **Policy Versions** screen displays:

- **Selected Version** – The currently selected version is marked **Selected**.
- **Create Time** – The day and time that the version was created. The time stamp on version #1 is the time when the original policy period was created. The time stamps on version #2 and #3 are the times when the **Side-by-Side** button was clicked. There is a small possibility that the time stamps for version #2 and #3 have different minute values because PolicyCenter creates the side-by-side periods sequentially.
- **Version Status** – The status of the version. For example, **Draft** or **Quoted**.
- **Premium Totals** – The value of the premium if the policy has been quoted.

Working with Side-by-side Quoting

This topic provides step-by-step instructions for working with side-by-side quoting in PolicyCenter.

- “Selecting Side-by-side Quoting in a Submission” on page 166
- “Editing a Version in a Policy Transaction with Side-by-side Quoting” on page 166
- “Binding and Issuing a Side-by-side Submission” on page 167

Note: The instructions are written for the personal auto line of business.

See also

- “Side-by-side Quoting Screen” on page 163.

Selecting Side-by-side Quoting in a Submission

Follow these steps to convert a submission policy transaction to side-by-side quoting and make changes in the side-by-side data.

Note: These instructions walk you through the steps in a **Quick Quote**. You can also select **Full Application** as the **Quote Type**.

1. Start a submission for personal auto.
2. On the **New Submissions** screen, select **Quick Quote** for **Quote Type**.
3. Add a driver and a vehicle.
4. Select **Versions** → **Start Side-by-Side**.

PolicyCenter displays the **Side-by-Side Quoting** screen. Because you did not select an offering, PolicyCenter applies the **Basic Program**, **Standard Program**, and **Premium Program** offerings to each side-by-side version, respectively. Each side-by-side period has been quoted. **Policy Premium** displays the value of the quote.

5. Compare the values for each **Policy Premium**.
6. Make changes to the coverages for one or more versions.
The coverages are side-by-side data that apply to each version. After you make changes, the value for **Policy Premium** disappears in all changed versions.
7. Click **Quote All** to regenerate the **Policy Premium**.
8. Click **Quote All** to generate quotes for all versions.

Editing a Version in a Policy Transaction with Side-by-side Quoting

Follow these steps to make changes to a version in a policy transaction with side-by-side quoting. If you change the base data, that change is copied to the other side-by-side versions. If you change side-by-side data, that change applies to the selected version only. These instructions assume that you are in a quick quote for a personal auto submission.

Note: In some circumstances, certain users cannot modify any versions in a side-by-side policy transaction. For more information, see “Side-by-side Versions Locked” on page 163.

1. In a policy transaction with side-by-side quoting, click **Tools** → **Side-by-Side Quoting** to jump to that screen.
Under the **Actions** menu, the version selector drop-down menu displays the names of the side-by-side versions.
 2. Select **Version #2** from the version selector.
 3. Click **Quick Quote Information** in the left sidebar. PolicyCenter displays this screen.
 4. Click **Edit Policy Transaction** to make the policy editable.
 5. In **Policy Info**, make a change to the **Term Type**. **Term Type** is a base data field.
 6. Click **Vehicles** → **New Vehicle**.
 7. Add a vehicle. A vehicle is side-by-side data. Therefore, the vehicle is only on **Version #2** the policy.
 8. Click **Tools** → **Side-by-Side Quoting**.
- Notice that all versions no longer have a **Policy Premium** value. The premium for the versions must be updated to reflect the new term type.
- The vehicle that you added is not on other versions of the policy.

9. In the version selector under the **Actions** menu, select **Version #1**.

10. Click **Back** to go to the **Quick Quote Information** screen for **Version #1**.

Notice that the **Term Type** has the new value that you set in step 5 on **Version #2**. Because term type is base data, PolicyCenter copies the value to the other versions.

Notice that the vehicle that you added to the other version does not appear. Because the vehicle is side-by-side data, PolicyCenter does not copy it to the other versions.

11. Click **Versions** → **View Side-by-Side Versions**.

12. Click **Quote All** to generate quotes for all versions.

Binding and Issuing a Side-by-side Submission

Follow these steps to bind and issue a submission with side-by-side quoting. These instructions assume that you are in a personal auto submission.

1. In a policy transaction with side-by-side quoting, click **Tools** → **Side-by-Side Quoting** to jump to that screen.

2. Click **Select** in the version that you want to bind and issue.

PolicyCenter displays the **Quote** page for the selected version.

3. If you are in **Quick Quote**, click **Full App**. You cannot bind and issue a quick quote submission. You may have to add additional information required for quoting. When you make the change to **Full App**, PolicyCenter invalidates the quotes and sets the policy periods back to draft status.

4. Click **Quote**.

5. Select **Bind Only** or **Issue Policy** from the **Bind Options** menu.

Multi-version Quoting

With multi-version quoting, you can generate multiple versions of a policy for comparison in a submission, renewal, and policy change policy transaction. You can select to view each version of the policy, and modify the coverages, terms and other parts of the policy. You can compare the status and premiums for all versions. In the default configuration, multi-version quoting is available in all line of business except for personal auto.

This topic includes:

- “Multi-version Quoting and Side-by-side Quoting” on page 169
- “Working with Multi-version Quoting” on page 169
- “Setting the Maximum Number of Multi-version Quotes” on page 170

Multi-version Quoting and Side-by-side Quoting

Multi-version quoting is similar to side-by-side quoting. In the default configuration, multi-version quoting is available in all lines of business except for personal auto. In the default configuration, side-by-side quoting is only available in the personal auto line of business. With multi-version quoting, you view and modify each version individually. In side-by-side quoting, you view and make changes to multiple versions in a screen that displays the versions next to each other.

See also

- “Side-by-side Quoting” on page 159
- “Side-by-side Quoting Compared with Multi-Version Quoting” on page 159

Working with Multi-version Quoting

This topic describes how to work with multi-version quoting in the PolicyCenter user interface.

Creating and Comparing Multi-version Quotes

You can create alternate versions that better reflect an applicant's requirements, and then compare them. You can create multiple versions in submission, renewal, and policy change policy transactions. You can create multiple versions after the policy is quoted.

1. In the Quote screen, click **Versions** → **Start Multi-Version**.

PolicyCenter creates a new version of the policy transaction that contains the previously entered data.

2. Make desired changes and click **Quote**.

3. If you want to compare the submissions, click **Policy Versions** under the **Tools** menu.

The **Policy Versions** screen appears with a heading indicating that you are viewing multiple versions and not side-by-side quoting. You can:

- Select a version and make it the selected version.
- Select a version and rename it.
- Select a version and withdraw it.
- Select two versions and click **Diff** to compare the differences.

Setting the Maximum Number of Multi-version Quotes

In the default configuration, you can create three versions by using multi-version quoting. The maximum number of quotes is specified in `config.xml` in Studio. Edit this file to change the maximum number of multi-version quotes:

Parameter	Default value
<code>RenewalMaxQuotes</code>	3
<code>SubmissionMaxQuotes</code>	3
<code>PolicyChangeMaxQuotes</code>	3

See also

- “Multiple Revision Jobs and the Job Selected Branch Property” on page 554 in the *Configuration Guide*

part IV

Lines of Business



chapter 20

Line of Business Overview

In the base configuration, PolicyCenter includes several common lines of business which are described in this part of the *Application Guide*. Each line of business contains a reference implementation that you can use to accelerate your implementation. Each line of business includes reference implementations for policy transactions, policy file screens, sample rating rules, sample eligibility and evaluation rules, and forms logic. The reference implementation also provides sample content for coverages, limits, deductibles, and other important data.

The lines of business in the base configuration are:

- Commercial Auto
- Businessowners
- Commercial Property
- Commercial Package Policy
- General Liability
- Inland Marine
- Personal Auto
- Workers' Compensation

This topic includes:

- “Additional Lines of Business” on page 173
- “Developing a New Line of Business” on page 174

Additional Lines of Business

In addition to the lines of business provided in the PolicyCenter base configuration, Guidewire provides additional line of business templates that are delivered as extension packs. These line of business templates offer a more complete set of content for the lines of business in certain locales. The line of business templates either add new lines of business or provide a more complete implementation of the lines of business in the base configuration. Some example line of business templates are:

- Homeowners
- Personal Umbrella
- Crime
- General Liability
- Commercial Property

- Commercial Auto
- Workers Compensation

Contact Guidewire Customer Support for more information about the line of business templates.

Developing a New Line of Business

You may need to support a line of business that is not in the PolicyCenter base configuration or in a line of business template extension pack. You can configure PolicyCenter to handle nearly any property and casualty insurance line of business. See “Adding a New Line of Business” on page 125 in the *Product Model Guide* for more information.

Commercial Auto

The base configuration of PolicyCenter provides a sample commercial auto policy implementation.

This line of business contains a reference implementation that you can use to accelerate your implementation. This line of business includes reference implementations for jobs, policy file screens, sample rating rules, sample eligibility rules, and forms logic. The reference implementation also provides sample content.

Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can build your own compliance system. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. The default application contains a sample set of these classifications.

This topic includes:

- “Commercial Auto Screens” on page 175
- “Commercial Auto Object Model” on page 187

Commercial Auto Screens

The commercial auto policy implementation contains a series of screens for pre-qualifying the applicant, adding locations and vehicles, assessing risk, quoting, and selecting payment options. This section provides descriptions of fields in the default configuration.

- “Offerings Screen for Commercial Auto” on page 176
- “Qualification Screen for Commercial Auto” on page 176
- “Policy Info Screen for Commercial Auto” on page 176
- “Locations Screen for Commercial Auto” on page 178
- “Vehicles Screen for Commercial Auto” on page 179
- “Commercial Auto Line Screen for Commercial Auto” on page 180
- “State Info Screen for Commercial Auto” on page 181
- “Drivers Screen for Commercial Auto” on page 182

- “Covered Vehicles Screen for Commercial Auto” on page 182
- “Modifiers Screen for Commercial Auto” on page 183
- “Risk Analysis Screen for Commercial Auto” on page 183
- “Policy Review Screen for Commercial Auto” on page 184
- “Quote Screen for Commercial Auto” on page 184
- “Forms Screen for Commercial Auto” on page 185
- “Payment Screen for Commercial Auto” on page 185

Note: The Commercial Auto line of business was originally named Business Auto. Therefore, many entity names and other internal PolicyCenter designations include the prefix **BusinessAuto** or **BA**.

Offerings Screen for Commercial Auto

The Offerings screen enables you to select an offering. In the default configuration, you can chose between **Standard** and **Special Risk**.

Configuring Commercial Auto Offerings

To determine the differences between the standard and special risk offerings, use Product Designer to navigate to **Product Model** → **Products** → **Commercial Auto**. Select **Offerings** to specify the offerings for each coverage, exclusion, condition, or modifier. You can change, add, and remove offerings using the controls in this screen.

To determine the differences between the **Standard** and **Special Risk** offerings, select the **Selections** subtab in the **Offerings** tab. Note the **Yes** and **(Modified Children)** entries in the **Modified** column of the **Special Risk (BASpecialRisk)** offering.

See also

- “Configuring Offerings” on page 99 in the *Product Model Guide*

Qualification Screen for Commercial Auto

The Qualification screen contains a set of questions to pre-qualify the applicant.

The questions reflect risks that the carrier wants to assess at the beginning of the submission. Since this is the gateway to the balance of the submission, the **intent is to determine eligibility of the applicant**.

The questions do not contain any regulatory requirements. In your implementation, the question set can impact later functionality and contain regulatory requirements.

The pre-qualification questions for commercial auto are **available in Submission and Rewrite New Account** policy transaction types in both offerings. In the default implementation, one question, **Asbestos**, blocks quoting when answered incorrectly. In Studio you can specify the correct answer and whether or not to raise an underwriting issue.

Configuring Commercial Auto Questions

Questions are defined in Product Designer in **Product Model** → **Question Sets** → **Commercial Auto Pre-Qualification**.

For information on configuring question sets, adding and removing questions, and specifying correct answers, failure messages, and blocking actions, see “**Question Sets**” on page 57 in the *Product Model Guide*.

Policy Info Screen for Commercial Auto

The **Policy Info** screen captures basic information about the policy.

This screen contains information about the primary named insured, producer information, policy details, and selected underwriting company. The buttons and choices that you see on the **Policy Info** screen vary based on the

contacts associated with the account and how they are associated with the policy. For example, under **Primary Named Insured** you might see any or all of the following:

- New Company
- New Person
- From Address Book
- Existing Contact

Not all choices appear at any given time.

The following table describes the key fields in the **Policy Info** screen.

Field name	Description
Primary Named Insured	<p>PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select:</p> <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact <p>If the named insured is a person, the social security number is required in Official IDs. If the named insured is a business, then FEIN is required.</p> <p>For Existing Contact, PolicyCenter lists contacts with the Account Holder or Named Insured role on the account. The list does not include contacts who are already the primary or additional named insureds on this policy.</p>
Organization Type	Enables you to select the type of organization such as whether this business is a corporation or partnership.
Additional Named Insureds	<p>Use to create additional named insureds. This might be a business partner, for example. The Add button enables you to select:</p> <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Additional Named Insured • Other Contacts
Additional Insureds	<p>Enables you to extend coverage to additional persons or companies based on the Type field. The Type field is required. You can specify a Type of Lessors. This extends coverage to persons or companies who lease vehicles to you.</p>
Policy Details	<p>Includes Term Type, Term Number, Effective Date, Expiration Date, Written Date, Rate as of Date, Fleet, Policy Type, and the Base State.</p> <p>Fleet is a required field that specifies whether the policy is a fleet or non-fleet policy. Values are:</p> <ul style="list-style-type: none"> • 10 or more units (Fleet policy) • Fewer than 10 units (Non-fleet policy) <p>You cannot change the fleet indicator in a policy change. To change between a fleet and non-fleet in an existing policy, you must cancel and rewrite the policy.</p> <p>Policy Type is a required field that specifies the type of commercial auto policy and its coverage form. In the U.S., there are currently four mutually-exclusive commercial auto coverage forms. In PolicyCenter, these coverage forms are implemented as <i>policy types</i>. Values are:</p> <ul style="list-style-type: none"> • Business Auto – Fleet or non-fleet coverages for vehicles used for business purposes. • Garagekeepers – Coverages for vehicles left in the care of the insured for service, repair, storage, or safekeeping, as well as vehicles held for sale with dealers or non-dealers. • Motor Carrier and Truckers – Coverages pertaining to the operation of trucks, trailers, and related equipment. • Business Auto Physical Damage – Coverages that include only physical damage to vehicles used for business purposes. <p>The Base State field is automatically set to the state of the policy address. You can change this value.</p>

Field name	Description
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p> <p>If you change the producer in the middle of a policy period, the new producer is the Producer of Service. The original Producer of Record remains. When the policy is renewed, the Producer of Service becomes the Producer of Record.</p>
Underwriting Companies	<p>The underwriting company is automatically set for a policy version based on logic defined in segmentation classes. You can select a different underwriting company if you have the correct permissions.</p> <p>The drop-down list contains a configurable set of choices.</p>
Preferred Currency	If PolicyCenter is configured as a multicurrency system, the Policy Info screen displays this label and the following fields related to the preferred currency.
Coverage	<p>The preferred or default currency for coverages on the policy. The Coverage currency choices come from the policy line configuration in Product Designer.</p> <p>In the base configuration, the default is the preferred coverage currency on the account, Account.PreferredCoverageCurrency.</p>
Settlement	<p>The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the Quote and other screens. In the base configuration you can select one of the following currencies:</p> <ul style="list-style-type: none"> • USD • EUR • GBP • CAD • AUD • RUB • JPY <p>The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, Account.PreferredSettlementCurrency. You can set the preferred settlement currency in the Account File Summary screen Currencies → Settlement field.</p>

Configuring the Policy Info Screen

You can configure the Policy Info screen in Studio by navigating to SubmissionWizard_PolicyInfoDV.pcf.

Locations Screen for Commercial Auto

In the Locations screen, enter the locations for vehicles on the policy.

By default, the primary location defaults to the primary location on the account. Add other locations that are covered by this policy. If you create a new location, PolicyCenter adds it to the account. Select a location and click Set As Primary to change the primary location.

In commercial auto policies, locations have a required Commercial Auto Line Territory Code field that corresponds to the address of the location.

For more information about locations, see “Locations” on page 343.

Configuring the Locations Screen

You can configure the Locations screen by using Studio to navigate to LocationsScreen.pcf.

You can configure territory codes in the territory_codes.xml system table in Product Designer.

See also

- “Locations” on page 343

Vehicles Screen for Commercial Auto

In the Vehicles screen, you create the vehicle or vehicles that this policy insures. Select the garage location and provide basic information such as VIN and cost for each covered vehicle.

The Vehicles screen has three buttons, **Create Vehicle**, **Remove Vehicle**, and **Clone Selected**. Clicking **Create Vehicle** displays three cards: **Vehicle Details**, **Coverages**, and **Additional Coverages**.

Vehicle Details Tab

Enter basic information about the vehicle in the **Vehicle Details** card.

Field name	Description
Where Garaged → Location Name	Choose the garage location of the vehicle. The drop-down list contains policy locations and account locations. Click the down arrow button to the right of the list to: <ul style="list-style-type: none">Edit the current garage locationCreate a new garage location Changing the location updates the account location, but does not change locations on other in-force policies.
Vehicle Type	Select a vehicle type. The default configuration includes: <ul style="list-style-type: none">Trucks, Tractors, TrailersPassenger VehiclesLivery VehiclesSpecial
VIN	Enter the vehicle identification number. This is an integration point. In the development environment, PolicyCenter supplies a demonstration plugin. In a production environment, this field would likely link to a working plugin that retrieves vehicle data based on the VIN number. See “PolicyCenter Integration Points” on page 34 for additional information.
Vehicle Condition When Purchased	Choose whether the vehicle is new or used.
Cost	The cost of the vehicle.
Class	Specify a class code for this vehicle type and fleet. Enter an appropriate class code or click the search icon to search for a class code that corresponds to the size of vehicle, primary use, and driving radius.
Business Vehicle Additional Interests	Add additional interests.
Vehicle Rate Modifiers	Select applicable vehicle rate modifiers when available. Available modifiers vary depending on the selected vehicle type and garage location.

Configuring the Vehicles Details Tab

You can configure the cards of the **Vehicle Details** screen by using Studio to navigate to the `BAVehiclePopup.pcf` file.

To configure the vehicle rate modifiers, navigate in Product Designer to **Product Model** → **Policy Lines** → **Commercial Auto Line** and select **Modifiers**. Among the modifiers listed, some appear as vehicle rate modifiers, some as state rating modifiers, and some as overall modifiers on the **Modifiers** screen.

Coverages Tab for Vehicles

Specify standard vehicle coverages and coverage terms on the **Coverages** tab. The **Coverages** tab displays coverages in the following category:

- Commercial Auto Owned Auto Physical Damage Group

Configuring Vehicle Coverages

You can configure coverages by navigating in Product Designer to **Product Model** → **Policy Line** → **Commercial Auto Line** and selecting **Coverages**. To enable a coverage to display on this tab in PolicyCenter, select one of the **Category** types listed above.

Additional Coverages Tab for Vehicles

Specify non-standard vehicle coverages and coverage term settings on the **Additional Coverages** tab. The **Additional Coverages** tab displays coverages in the following categories:

- Commercial Auto Audio Visual Data Equipment Group
- Commercial Auto Tape Disc Record Group
- Commercial Auto Rental Group
- Commercial Auto Loan Lease Gap Group

You can configure coverages by navigating in Product Designer to **Product Model** → **Policy Line** → **Commercial Auto Line** and selecting **Coverages**. To enable a coverage to display on this tab in PolicyCenter, select one of the **Category** types listed above.

Commercial Auto Line Screen for Commercial Auto

On the **Commercial Auto Line** screen, you can add standard and additional coverages as well as exclusions and conditions.

When multicurrency display is enabled, you can set the currency on a coverable in the user interface. PolicyCenter then copies this coverage currency to each coverage on the coverable. The main screen for each coverable has the following multicurrency field:

Field	Description
Coverages in	Select the currency for the coverages on this coverable. By default, this value is set to the Policy Info → Preferred Currency → Coverage field. The drop-down list contains the Available Coverage Currencies defined in Product Designer on the main screen of the policy line.

See also

- “Multicurrency Policies” on page 517

Coverages Tab for Commercial Auto Line

The **Coverages** tab enables you to add line-level coverages in the following categories:

- Commercial Auto Owned Liability Group
- Commercial Auto Hired Auto Group
- Commercial Auto Non-Owned Group

Note: If you select any hired auto coverages, you must select at least one hired auto state. For each hired auto state you add, you must either specify the cost of hire for that state or select “If Any.” Similarly, if you select any non-owned coverage, you must select at least one non-owned liability state. For each non-owned liability state you add, you must specify the number of employees, total partners, and total volunteers.

Configuring the Coverages Tab

You can configure coverages by navigating in Product Designer to **Product Model** → **Policy Line** → **Commercial Auto Line** and selecting **Coverages**. To enable a coverage to display on this tab in PolicyCenter, select one of the **Category** types listed above.

Additional Coverages Tab for Commercial Auto Line

The **Additional Coverages** tab enables you to add the coverages in the following category:

- **Commercial Auto Drive Other Car Group**

Configuring the Additional Coverages Tab

You can configure coverages by navigating in Product Designer to **Product Model** → **Policy Line** → **Commercial Auto Line** and selecting **Coverages**. To enable a coverage to display on this tab in PolicyCenter, select one of the **Category** types listed above.

Exclusions & Conditions Tab for Commercial Auto Line

The **Exclusions & Conditions** tab enables you to enter line-level exclusions and conditions. In the default configuration, no exclusions or conditions are configured at the line level.

Configuring the Exclusions & Conditions Tab

To add line-level exclusions and conditions, navigate in Product Designer to **Product Model** → **Policy Line** → **Commercial Auto Line**. Select **Exclusions** or **Conditions** to add new exclusions or conditions.

State Info Screen for Commercial Auto

For each jurisdiction on the policy that is used as a garaging location, the **State Info** screen enables you to specify coverages, exclusions, conditions, and jurisdictional rating modifiers.

Coverages Tab for States

For each jurisdiction that is a garage location on the policy, the **Coverages** tab on the **State Info** screen enables you to add coverages in the following categories:

- **Commercial Auto Owned Vehicle Group by State**
- **Commercial Auto PIP Coverages** – This category only appears if the vehicle is garaged in a jurisdiction that offers PIP coverages.

Configuring the Coverages Tab for States

You can configure coverages by navigating in Product Designer to **Product Model** → **Policy Line** → **Commercial Auto Line** and selecting **Coverages**. To enable a coverage to display on this tab in PolicyCenter, select one of the **Category** types listed above.

You can configure PIP availability by selecting the PIP coverage for a particular state and clicking the **Availability** subtab.

Additional Coverages Tab for States

The **Additional Coverages** tab on the **State Info** screen enables you to add the coverages in the following categories:

- **Commercial Auto Pollution Group**
- **Commercial Auto Fellow Employees Group**
- **Commercial Auto Loss of Use Group**
- **Non-owned Social Services Extended** – Only appears if the non-owned coverages were selected in the **Commercial Auto Line** screen for one or more vehicles garaged in the selected state.

Configuring the Additional Coverages Tab for States

You can configure coverages by navigating in Product Designer to **Product Model** → **Policy Line** → **Commercial Auto Line** and selecting **Coverages**. To enable a coverage to display on this tab in PolicyCenter, select one of the **Category** types listed above.

Exclusions & Conditions Tab for States

The **Exclusions & Conditions** tab enables you to add exclusions and conditions for each jurisdiction on the policy in the following category:

- Commercial Auto Terrorism Group

Configuring the Exclusions & Conditions Tab for States

You can configure coverages by navigating in Product Designer to **Product Model** → **Policy Line** → **Commercial Auto Line** and selecting **Coverages**. To enable a coverage to display on this tab in PolicyCenter, select one of the **Category** types listed above.

State Rating Tab for States

The **State Rating** tab enables you to enter modifier values and credits and debits for each jurisdiction on the policy.

A modifier is a value used by the rating engine to adjust the policy premium or some portion of the premium. Modifiers capture information relevant to the pricing of a policy that are not necessarily tied to a specific coverable or coverage.

In the default configuration, the commercial auto line has an experience modifier and an expense modifier. Validation on this screen prevents you from specifying a value of less than 0.5 or greater than 5.

Configuring the Modifiers on the State Rating Tab

To configure the state rating modifiers, navigate to **Product Model** → **Policy Lines** → **Commercial Auto Line** and select **Modifiers**. Among the modifiers listed, some appear as vehicle rate modifiers, some as state rating modifiers, and some as overall modifiers on the **Modifiers** screen.

Drivers Screen for Commercial Auto

Add drivers covered by the policy on the **Drivers** screen. You must manually enter the information for each driver. You cannot select account contacts on the **Drivers** screen. For each driver, you must specify:

- First Name
- Last Name
- Date of Birth
- License #
- License State

Configuring the Drivers Screens

You can configure the **Drivers** screen by navigating to **BADriverScreen.pcf** in Studio.

You can configure the **Driver Details** screen by navigating to **BADriverPopup.pcf**.

Covered Vehicles Screen for Commercial Auto

The **Covered Vehicles** screen displays the vehicle groups for the current policy. Each row corresponds to a coverage such as **Liability** or **PIP**. The columns represent a vehicle group such as **OVO** (owned vehicles only) or **DVO** (designated vehicles only). For each coverage, the word **Yes** appears in the vehicle group column if the policy contains coverages for that group. The system sets these values based on selected coverages and their corresponding

coverage symbols. However, the values can be manually overridden by a user with appropriate authority (**Edit covered auto symbols** permission).

To override coverage symbols, click **Edit Covered Vehicles**, then add or remove coverage symbols by selecting or clearing the corresponding check boxes. Choosing to edit covered vehicles permanently changes the policy so that PolicyCenter can no longer update the information on this screen. Therefore, after you edit the symbols, the screen displays the following message in red: **Covered autos were manually edited.** From this point forward, changes to the policy do not overwrite the manual selections.

Configuring the Covered Vehicles Screen and Coverage Symbol Groups

You can configure the **Covered Vehicles** screen by navigating to `CoveredAutoSymbolsScreen.pcf` in Studio.

To configure coverage symbol groups for commercial auto, in Product Designer navigate to **Products** → **Policy Lines** → **Commercial Auto Line** and click **Coverage Symbol Groups**. Click a coverage symbol group to display the **Coverage Symbol Patterns**. The **Coverage Symbol Patterns** shows the symbol patterns (referred to as vehicle groups in PolicyCenter) defined for that coverage symbol group. You can add or remove coverage symbol patterns and change the symbol types that belong to each group.

The `BusinessAutoLineEnhancement.gsx` Gosu file contains the method `setCoveredAutoSymbols` that controls whether `Yes` initially appears in a column for a vehicle group.

Modifiers Screen for Commercial Auto

You can use the **Modifiers** screen to apply credits or debits that affect rating. In the default configuration, three modifiers are available:

- Location Dispersion Credit
- Liability Rates Modifier, with four categories of credits/debits
- Physical Damage Rates Modifier, with five categories of credits/debits

Each of the individual credit/debit categories has a specified minimum and maximum value, and the modifier has an overall minimum and maximum value. Validation ensures that you do not exceed the individual values. The overall values are enforced by code that prevents them from exceeding their maximum or minimum specified range, so no validation is needed.

Configuring the Modifiers on the Modifiers Screen

To configure the rating modifiers, navigate to **Product Model** → **Policy Lines** → **Commercial Auto Line** and select **Modifiers**. Among the modifiers listed, some appear as vehicle rate modifiers, some as state rating modifiers, and some as overall modifiers on the **Modifiers** screen. Select a modifier in the list to configure the properties of the modifier.

Risk Analysis Screen for Commercial Auto

The **Risk Analysis** screen enables you to enter data that the rating engine uses to evaluate the risk of the applicant.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – This tab appears when viewing a policy. It does not appear in a work order such as a submission. This tab allows you to add an underwriting referral reason to a policy. Underwriting referral reasons are usually only used when a notable condition arises outside of a job, possibly outside of the data that PolicyCenter maintains on the policy. For more information about this tab, see “Working with Underwriting Referral Reasons” on page 416.
- **UW Issues** – This tab displays underwriting issues that were raised during the job. For more information on this tab, see “Working with Underwriting Issues” on page 414.
- **Prior Policies** – This tab displays information about prior policies usually with another carrier. You can add information about prior policies.

- **Claims** – This tab allows you to search for loss claims on a related policy or by loss date. When integrated with a claim system, this tab displays claims from the claim system. For more information, see “Viewing Loss Claims for Policies” on page 716.
- **Prior Losses** – This tab displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

Policy Review Screen for Commercial Auto

The Policy Review screen is similar to the screen in other lines of business. For commercial auto, you can review coverages by jurisdiction.

For each jurisdiction, click the jurisdiction name to jump to the **State Location Summary**. This screen displays the locations in that jurisdiction. Each location lists coverage information about each vehicle garaged at that location.

For a submission job, this screen contains all the policy data in summary form. For other jobs, this screen displays the differences between the policy versions. It is useful to review this screen before generating a quote. If any of the information needs to be changed, click the menu link on the left sidebar to edit the appropriate screen.

If PolicyCenter is configured as a multicurrency system, the Policy Review screen displays the cost for each coverable’s coverages in the currency set on the coverable.

Configuring the Policy Review Screen

You can configure the Policy Review tab by navigating in Studio to `PolicyLineSummaryPanelSet.BusinessAutoLine.pcf`.

You can configure the State Location Summary screen by navigating in Studio to `BALocationSummaryPopup.pcf`.

The Differences tab is the standard modal panel set used throughout PolicyCenter. It can be configured by navigating to `PolicyChangeWizard.pcf` and selecting the BusinessAuto mode.

Quote Screen for Commercial Auto

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen.

This screen is similar to the quote screen for other lines of business. It contains the basic information at the top including the total premium and taxes and surcharges. The bottom of the screen displays a breakdown of the premium.

Within the **Quote** screen, your choices at this point include:

- Editing the policy (and thereby invalidating the current quote)
- Creating a new version of the policy
- Saving a draft of the policy
- Binding or issuing the policy
- Withdrawing, declining, or not taking the policy
- Printing the policy

PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

In addition, Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

If you have Guidewire Rating Management, the **Policy Premium** tab has a **Show Rate Worksheet** button. For more information about this button, see “Rating Worksheets” on page 548 in the *Application Guide*.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Forms Screen for Commercial Auto

The **Forms** screen allows you to associate forms with a policy.

PolicyCenter does not store the content of any forms associated with a policy. Therefore, the **Forms** screen only identifies that zero, one, or more, forms have been associated with the policy. The **Forms** screen simply lists form instances that indicate the physical forms, which are usually printed by an issuance system. Each form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form. The form content is not stored within PolicyCenter.

When you click **Quote** or **Bind**, PolicyCenter infers which forms the policy needs. In the default application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy Forms” on page 461
- “Policy Form Pattern Administration” on page 683 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment Screen for Commercial Auto

The **Payment** screen displays payment information for the policy.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- **Total Premium**
- **Taxes and fees**

- **Total Cost** – The sum of the previous fields.

Billing

Under the **Billing** heading, you can select:

- **Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- **Alt Billing Account** – Select an alternate billing account.
- **Billing Contact** – Select a person or company as the billing contact.

Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- **List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer's mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample billing accounts. If you are integrated with BillingCenter, the plugin returns a list of billing accounts retrieved from BillingCenter.

- **Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent's account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle's account as the alternate billing account. The uncle's account is not a subaccount of the parent's account.

You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.

- **Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample installments plans. If you are integrated with BillingCenter, the plugin returns a list of valid installment plans for the selected list bill payer. The integration retrieves installments plans from BillingCenter.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the **StandAloneBillingSystemPlugin** or the default integration with BillingCenter, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The **StandAloneBillingSystemPlugin** simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payments system, “**Payment Integration**” on page 553 in the *Integration Guide*.

If you select **Automatic** for **Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.
- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

Note: The BillingCenter integration does not support this functionality.

Commercial Auto Object Model

This section describes the objects or entities associated with the commercial auto line of business and contains the following sections:

- “**Commercial Auto Object Model Overview**” on page 188
- “**Coverages in Commercial Auto**” on page 189
- “**Modifiers in Commercial Auto**” on page 189

See also

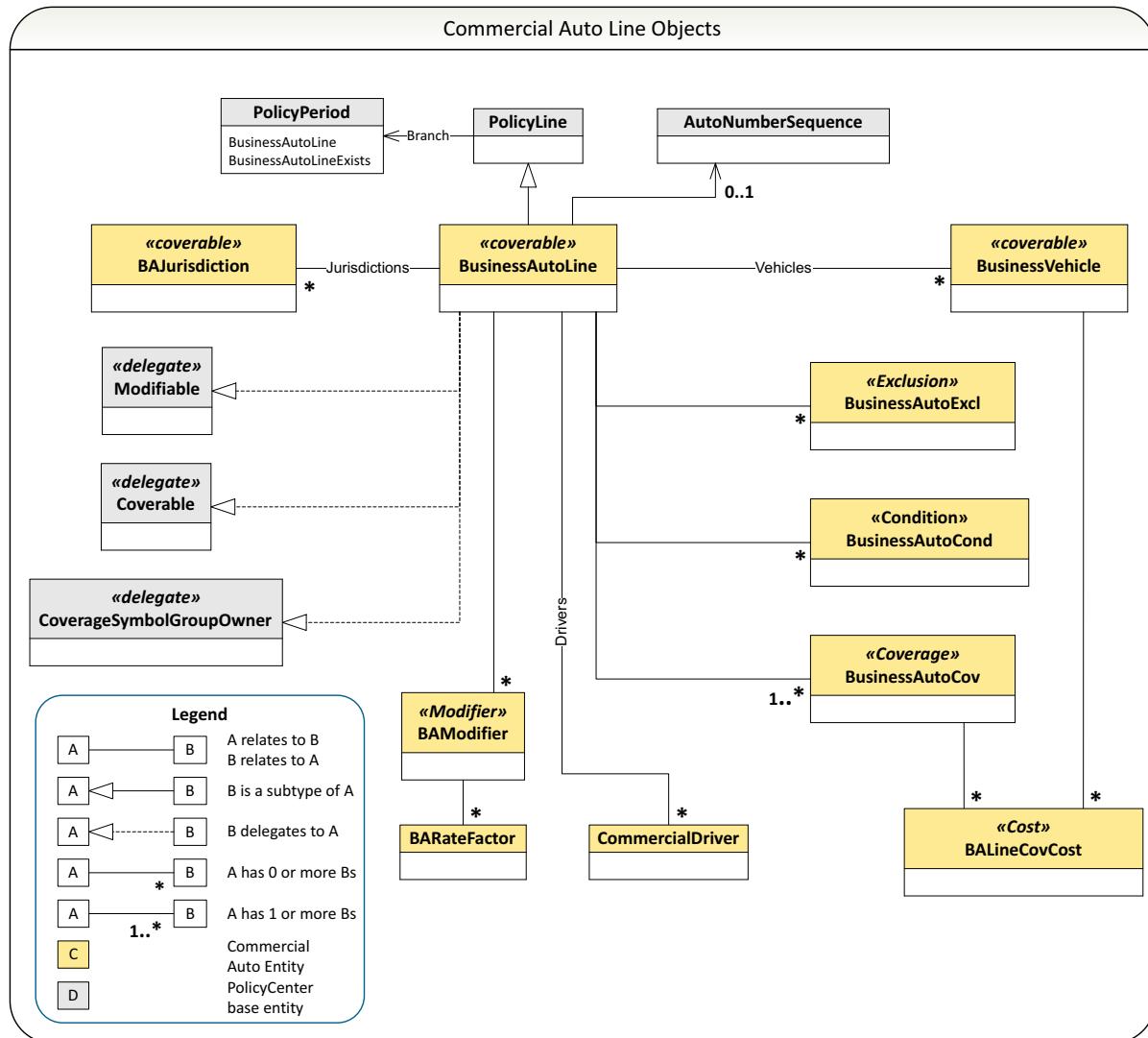
- “**Core Entities Associated with Policies**” on page 309
- “**Cost and Transaction Model for Business Auto Line**” on page 424

Commercial Auto Object Model Overview

This topic describes the object model for the commercial auto line.

PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.

The following object model diagram shows the main entities for the commercial auto line.



The object model diagram shows the relationships between the various entities associated with commercial auto policies. The `PolicyLine` entity contains subtypes for each line of business. One of these is `BusinessAutoLine`.

Note: This diagram shows a partial listing of entities in the commercial auto line. For the complete list of entities and properties, see the *Data Dictionary*. Be aware that the Commercial Auto line of business was originally named Business Auto. Therefore, entity, method, and property names, as well as other internal PolicyCenter designations, use the prefix `BusinessAuto` or `BA`.

Coverages in Commercial Auto

PolicyCenter defines a coverage as a protection from a specific risk. A coverage entity must implement the `Coverage` interface. Coverages always attach to a `Coverable`. There are two types of coverages: property and liability.

In the default configuration, the commercial auto policy line contains the following types of coverages:

Coverage type	Attaches to	Description
<code>BusinessAutoCov</code>	<code>BusinessAutoLine</code>	Coverage choices that apply to the entire policy.
<code>BAStateCov</code>	<code>BAJurisdiction</code>	Coverages that apply to jurisdictions. This coverage has one subtype: <ul style="list-style-type: none">• <code>BAHiredSpecPerilCov</code>
<code>BusinessVehicleCov</code>	<code>BusinessVehicle</code>	Coverages for vehicles. This coverage has one subtype: <ul style="list-style-type: none">• <code>BASpecCausesLossCov</code>

Modifiers in Commercial Auto

A modifier is a value used by the rating engine to adjust the policy premium or some portion of the premium. A modifier captures information relevant to the pricing of a policy that is not necessarily tied to a specific coverable or coverage. In the commercial auto line, the `BAModifier` entity represents modifiers for this policy line.

Locations in Commercial Auto

The commercial auto line does not define its own entity for locations. Instead, the `BusinessVehicle` entity has a foreign key that points to a `PolicyLocation` entity which is the garage location of the vehicle. The `TerritoryCode` on `PolicyLocation` is used for rating.

Drivers in Commercial Auto

The `CommercialDriver` entity is accessed through an array from `BusinessAutoLine`. The `CommercialDriver` entity represents a driver on the policy. In commercial auto, drivers are not contacts on the account. Therefore, unlike personal auto, the `CommercialDriver` entity is not a subtype of the `Contact` entity.

Jurisdictions in Commercial Auto

The `BAJurisdiction` entity is accessed through an array from `BusinessAutoLine`. The `State` field is a typekey to the jurisdiction that is covered. The `BAJurisdiction` entity has an array to access the coverages for the jurisdiction.

Businessowners

The base configuration of PolicyCenter provides a sample businessowners policy implementation.

This line of business contains a reference implementation that you can use to accelerate your implementation. This line of business includes reference implementations for jobs, policy file screens, sample rating rules, sample eligibility rules, and forms logic. The reference implementation also provides sample content.

Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can build your own compliance system. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. The default application contains a sample set of these classifications.

This topic includes:

- “Businessowners Screens” on page 191
- “Businessowners Object Model” on page 200

Businessowners Screens

The businessowners policy implementation contains a series of screens for pre-qualifying the applicant, describing locations and buildings, choosing coverages, assessing risk, quoting, and selecting payment options. This section provides descriptions of fields in the default configuration.

- “Offerings Screen for Businessowners” on page 192
- “Qualification Screen for Businessowners” on page 192
- “Policy Info Screen for Businessowners” on page 192
- “Businessowners Line Screen” on page 194
- “Locations Screen for Businessowners” on page 195
- “Buildings and Locations Screen for Businessowners” on page 196
- “Modifiers Screen for Businessowners” on page 197
- “Risk Analysis Screen for Businessowners” on page 197

- “Policy Review Screen for Businessowners” on page 197
- “Quote Screen for Businessowners” on page 197
- “Forms Screen for Businessowners” on page 198
- “Payment Screen for Businessowners” on page 198

Offerings Screen for Businessowners

The **Offerings** screen contains a set of questions related to offerings.

Offerings let you define different product types for different types of buyers. Answers to the questions can affect which offerings are available. Offerings can filter parts of the product model such as policy terms, policy lines in a package policy, coverages, covterms, covterm options and packages, modifiers, and question sets.

In the default implementation, the answers to the questions determine the choices on the **Offering Selection** drop-down list.

For more information about offerings, see “Understanding Offerings” on page 477.

Qualification Screen for Businessowners

This screen contains a set of questions to pre-qualify the applicant.

The questions reflect risks that the carrier wants to assess at the beginning of the submission. Since this is the gateway to the balance of the submission, the intent is to determine eligibility of the applicant.

In the default implementation, the question set does not impact functionality in any other part of the application. The questions do not contain any regulatory requirements. In your implementation, the question set can raise underwriting issues, impact later functionality, and contain regulatory requirements.

Policy Info Screen for Businessowners

The **Policy Info** screen captures basic information about the policy.

This screen contains information about the primary named insured, producer information, policy details, and selected underwriting company. The buttons and choices that you see on the **Policy Info** screen vary based on the contacts associated with the account and how they are associated with the policy. For example, under **Primary Named Insured** you might see any or all of the following:

- New Company
- New Person
- From Address Book
- Existing Contact

Not all choices appear at any given time.

The following table describes the key fields in the **Policy Info** screen.

Field name	Description
Primary Named Insured	<p>PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select:</p> <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact <p>If the named insured is a person, the social security number is required in Official IDs. If the named insured is a business, then FEIN is required.</p> <p>For Existing Contact, PolicyCenter lists contacts with the Account Holder or Named Insured role on the account. The list does not include contacts who are already the primary or additional named insureds on this policy.</p>
Business and Operations	Includes the year the business was started and a description of operations.
Organization Type	Enables you to select the type of organization such as whether this business is a corporation or partnership.
Additional Named Insureds	<p>Use to create additional named insureds. This might be a business partner, for example. The Add button enables you to select:</p> <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Additional Named Insured • Other Contacts
Policy Details	Includes the Term Type , Effective Date , Expiration Date , Written Date , and the Base State .
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p> <p>If you change the producer in the middle of a policy period, the new producer is the Producer of Service. The original Producer of Record remains. When the policy is renewed, the Producer of Service becomes the Producer of Record.</p>
Underwriting Companies	<p>The underwriting company is automatically set for a policy version based on logic defined in segmentation classes. You can select a different underwriting company if you have the correct permissions.</p> <p>The drop-down list contains a configurable set of choices.</p>
Preferred Currency	If PolicyCenter is configured as a multicurrency system, the Policy Info screen displays this label and the following fields related to the preferred currency.

Field name	Description
Coverage	<p>The preferred or default currency for coverages on the policy. The Coverage currency choices come from the policy line configuration in Product Designer.</p> <p>In the base configuration, the default is the preferred coverage currency on the account, Account.PreferredCoverageCurrency.</p>
Settlement	<p>The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the Quote and other screens. In the base configuration you can select one of the following currencies:</p> <ul style="list-style-type: none"> • USD • EUR • GBP • CAD • AUD • RUB • JPY <p>The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, Account.PreferredSettlementCurrency. You can set the preferred settlement currency in the Account File Summary screen Currencies → Settlement field.</p>

Businessowners Line Screen

The **BusinessOwners** screen allows you to enter coverages which apply to the entire policy. This screen has the following tabs:

- **Included Coverages** – Coverages that have been defined as **Required** or **Suggested**.
- **Additional Coverages** – Coverages that have not been defined as **Electable**. Select coverage options from a drop-down list.

A coverage can be configured in product model to be either **Required**, **Suggested**, or **Electable**. To learn more about configuring coverages, read about coverage existence in “Adding Coverages to a Policy Line” on page 24 in the *Product Model Guide*.

When multicurrency display is enabled, you can set the currency on a coverable in the user interface. PolicyCenter then copies this coverage currency to each coverage on the coverable. The main screen for each coverable has the following multicurrency field:

Field	Description
Coverages in	<p>Select the currency for the coverages on this coverable. By default, this value is set to the Policy Info → Preferred Currency → Coverage field. The drop-down list contains the Available Coverage Currencies defined in Product Designer on the main screen of the policy line.</p>

See also

- “Multicurrency Policies” on page 517

Included Coverages

The following table describes some of the key fields in the **Included Coverages** tab on the **BusinessOwners** screen.

Field name	Description
Small Business Type	In the default application, you must select a Small Business Type . If not selected, a validation warning appears. The value selected may affect the type coverages that are available to the insured. For example, if you select Contractor-artisan or Contractor-landscape , the Contractors Tools/Installation coverage is required and is added to the Liability Coverages .
Property Coverage → Blanket	You can select whether you would like a single blanket limit to be used for property and/or contents coverage for all buildings on the policy. For example, if you select blanket building coverage, the single blanket limit would be the sum of all building limits on the policy PolicyCenter stores these values in the BlanketType typelist which you can view in Studio by navigating to Typelists → BlanketType .
Policywide Property Deductible	These fields apply to all buildings on the policy. If there are many buildings, there is no option to use different deductible values.
Liability Coverages → Liability	Limits – Allows the you to select packages which are in ratios of 1:2:2 or 1:3:2. Deductibles – Select whether the deductible applies to each claim or each occurrence.
Liability Coverages → Tenant Fire Liability	PolicyCenter provides a basic limit by default. You can add additional limits to the default application.
Liability Coverages → Premises Medical Expense	This is a suggested coverage and is selected by default. Although it is usually included, it must be excluded in certain businesses such as swimming pool equipment retail. It may be desirable to exclude in certain other businesses such as exercise studios or martial arts dojos.
Liability Coverages → Personal & Advertising Injury	This coverage is selected by default. While it is usually included, it must be excluded in certain businesses such as law and labor union offices, guard and detective agencies, and advertising agencies.

The PCF file for the **Included Coverages** tab is **BOPScreen.pcf**.

Additional Coverages on Businessowners Line Screen

You can select **Add Coverages** to add other electable coverages to the policy.

The PCF file for the **Additional Coverages** tab is the **BOPScreen.pcf**.

Locations Screen for Businessowners

In the **Locations** screen, enter the locations that this policy covers.

By default, the primary location defaults to the primary location on the account. Add other locations that are covered by this policy. If you create a new location, PolicyCenter adds it to the account. Select a location and click **Set As Primary** to change the primary location.

In businessowners policies, locations have an additional **Public Protection** field.

For more information about locations, see “Locations” on page 343.

Additional Coverages

The **Location Add. Coverages** tab is similar to the “Additional Coverages on Businessowners Line Screen” on page 195. Coverages available on this tab apply only to the selected location.

Location Questions

This screen contains a question set which you can configure. You can enhance this page to present different question sets based on your business needs.

Buildings and Locations Screen for Businessowners

Buildings are the fundamental rating unit for businessowners policies. A single building class code applies to both property and liability coverages. Businessowners policies must contain both property and liability coverage. Every policy must have at least one location and building. Conversely, multiple buildings at a single location may be treated as a single building with blanketed values.

The **Buildings and Locations** screen displays all previously added locations and allows you to add or select buildings at the highlighted location.

When you select a location, PolicyCenter displays buildings at that site and allows you to add buildings to that site. Clicking the **Add** button opens a **Details** screen where you can edit the building details. Clicking on a existing building also opens this screen.

Building Details Tab

The **Building Details** tab allows you to enter building attributes that will be used for rating, insurance to value testing, and underwriting information.

Details Tab

The **Details** tab has the following key fields.

Field name	Description
Building Class Code	Enter a building class code. You cannot update a building without supplying a class code. The drop-down list may also be pre-filled with codes filtered by the industry code of the primary named insured on the policy.
Building coverage	This is a suggested coverage because it is not required for renters.
Premium Basis Type	The class code determines the value of this read-only field. The basis types are: <ul style="list-style-type: none"> • Liability Limit – Most stores and general business operations • Payroll – Contractors • Sales – Motels, restaurants, and some types of stores • Building Limit – Lessors Risk Only (LOR)
Premium Basis Amount	An input field which is only visible if the basis type is Sales or Payroll .
Business Personal Property	This coverage is commonly referred to as contents or inventory. The coverage is a suggested coverage because it may not be required if the owner of a building is leasing it to others.
Building Construction	Provides fields to collect information that PolicyCenter uses in rating and testing insurance to value.
Building Improvement	Enter year of building improvement in YYYY format. The fields may be null. Values must be between Building Construction → Year Built and the current date (inclusive).
Burglar Alarm	This data is collected for rating purposes. All fields may be null.
Exposure	These values provide underwriting information which is not directly used by rating, but may affect premium modification.
Interest/Occupied/Leased	These fields provide data for rating and forms inference.
Additional Interests	This is a listview that collects information related to people or companies who have an insurable interest in the building or contents such as a mortgage holder or lender.

Building Additional Coverages

This tab is similar to “Additional Coverages on Businessowners Line Screen” on page 195. Coverages on this tab apply only to the selected building.

Modifiers Screen for Businessowners

The **Modifiers** screen displays scheduled credits which are applied at the jurisdictional level because filings and regulatory limits on credits and debits are set by the jurisdictions.

Risk Analysis Screen for Businessowners

The **Risk Analysis** screen allows you to enter data that is used to evaluate the risk of the applicant.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – This tab appears when viewing a policy. It does not appear in a work order such as a submission. This tab allows you to add an underwriting referral reason to a policy. Underwriting referral reasons are usually only used when a notable condition arises outside of a job, possibly outside of the data that PolicyCenter maintains on the policy. For more information about this tab, see “Working with Underwriting Referral Reasons” on page 416.
- **UW Issues** – This tab displays underwriting issues that were raised during the job. For more information on this tab, see “Working with Underwriting Issues” on page 414.
- **Prior Policies** – This tab displays information about prior policies usually with another carrier. You can add information about prior policies.
- **Claims** – This tab allows you to search for loss claims on a related policy or by loss date. When integrated with a claim system, this tab displays claims from the claim system. For more information, see “Viewing Loss Claims for Policies” on page 716.
- **Prior Losses** – This tab displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

Policy Review Screen for Businessowners

The **Policy Review** screen is similar to the summary screen for other lines of business.

For a submission job, this screen contains all the policy data in summary form. For other jobs, this screen displays the differences between the policy versions. It is useful to review this screen before generating a quote. If any of the information needs to be changed, click the menu link on the left sidebar to edit the appropriate screen.

If PolicyCenter is configured as a multicurrency system, the **Policy Review** screen displays the cost for each coverable's coverages in the currency set on the coverable.

Quote Screen for Businessowners

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen.

This screen is similar to the quote screen for other lines of business. It contains the basic information at the top including the total premium and taxes and surcharges. The bottom of the screen displays a breakdown of the premium.

Within the **Quote** screen, your choices at this point include:

- Editing the policy (and thereby invalidating the current quote)
- Creating a new version of the policy
- Saving a draft of the policy
- Binding or issuing the policy
- Withdrawing, declining, or not taking the policy
- Printing the policy

PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

In addition, Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

If you have Guidewire Rating Management, the **Policy Premium** tab has a **Show Rate Worksheet** button. For more information about this button, see “Rating Worksheets” on page 548 in the *Application Guide*.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Forms Screen for Businessowners

The **Forms** screen allows you to associate forms with a policy.

PolicyCenter does not store the content of any forms associated with a policy. Therefore, the **Forms** screen only identifies that zero, one, or more, forms have been associated with the policy. The **Forms** screen simply lists form instances that indicate the physical forms, which are usually printed by an issuance system. Each form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form. The form content is not stored within PolicyCenter.

When you click **Quote** or **Bind**, PolicyCenter infers which forms the policy needs. In the default application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy Forms” on page 461
- “Policy Form Pattern Administration” on page 683 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment Screen for Businessowners

The **Payment** screen displays payment information for the policy.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- **Total Premium**
- **Taxes and fees**
- **Total Cost** – The sum of the previous fields.

Billing

Under the **Billing** heading, you can select:

- **Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- **Alt Billing Account** – Select an alternate billing account.
- **Billing Contact** – Select a person or company as the billing contact.

Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- **List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer's mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample billing accounts. If you are integrated with BillingCenter, the plugin returns a list of billing accounts retrieved from BillingCenter.

- **Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent's account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle's account as the alternate billing account. The uncle's account is not a subaccount of the parent's account.

You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.

- **Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the `StandAloneBillingSystemPlugin`, the plugin returns a list of sample installments plans. If you are integrated with BillingCenter, the plugin returns a list of valid installment plans for the selected list bill payer. The integration retrieves installments plans from BillingCenter.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the `StandAloneBillingSystemPlugin` or the default integration with BillingCenter, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The `StandAloneBillingSystemPlugin` simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payments system, “**Payment Integration**” on page 553 in the *Integration Guide*.

If you select **Automatic** for **Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.
- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

Note: The BillingCenter integration does not support this functionality.

Businessowners Object Model

This section describes the objects or entities associated with the businessowners line of business and contains the following sections:

- “**Businessowners Object Model Overview**” on page 201
- “**Coverages in the Businessowners**” on page 202

- “Modifiers in Businessowners” on page 202

See also

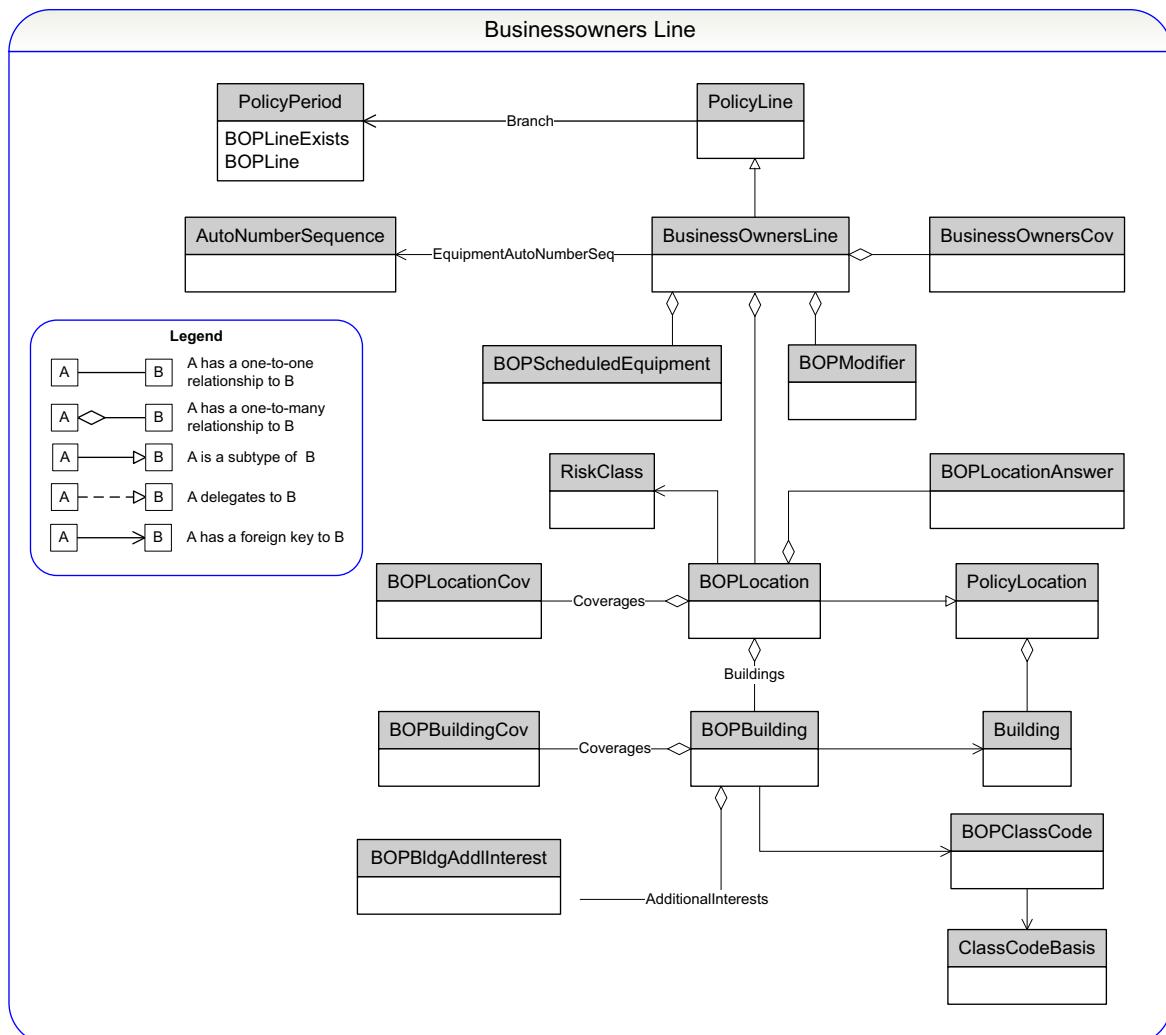
- “Core Entities Associated with Policies” on page 309
- “Cost and Transaction Model for Businessowners Line” on page 424

Businessowners Object Model Overview

This topic describes the object model for the businessowners line.

PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.

The following object model diagram shows the main entities for the businessowners line.



The object model diagram shows the relationships between the various entities associated with businessowners policies. The `PolicyLine` entity contains subtypes for each line of business. One of these is `BusinessOwnersLine`.

Note: This diagram shows a partial listing of entities in the businessowners line. For the complete list of entities and properties, see the *Data Dictionary*.

Coverages in the Businessowners

PolicyCenter defines a coverage as a protection from a specific risk. A coverage entity must implement the `Coverage` interface. Coverages always attach to a `Coverable`. There are two types of coverages: property and liability. For example, a businessowners policy provides coverage for buildings owned or leased by the business.

In the base configuration, the businessowners policy line contains the following types of coverages:

Coverage type	Attaches to	Description
<code>BusinessOwnersCov</code>	<code>BusinessOwnersLine</code>	Coverage choices that apply to the entire policy, such as <code>Policywide Property Deductible</code> .
<code>BOPLocationCov</code>	<code>BOPLocation</code>	Coverage choices that apply to a location, such as <code>Money and Securities Cov</code> .
<code>BOPBuildingCov</code>	<code>BOPBuilding</code>	Coverage choices that apply to a building, such as <code>Building Coverage</code> .

Modifiers in Businessowners

A modifier is a value used by the rating engine to adjust the policy premium or some portion of the premium. A modifier captures information relevant to the pricing of a policy that is not necessarily tied to a specific coverable or coverage. The businessowners line has the following modifier:

Modifier type	Applies to	Description
<code>BOPModifier</code>	<code>BusinessOwners Line</code>	A modifier of the policy line. In the default application, there is one modifier called <code>Schedule Credit</code> . This modifier contains an array of rate factors such as building features, employees, management, premises and equipment, protection, and risk elements not addressed in the classification plan.

Commercial Package Policy

Commercial package policy combines multiple lines of business into a single policy for easy administration and as a convenience to the insured. Each commercial package policy has a single policy number. Legally, each of the component lines is defined as a coverage part instead of as a separate policy. However, each component line of business is sufficiently defined to stand independently as a policy.

In the default configuration, the commercial package policy consists of the general liability, commercial property, and inland marine lines of business.

This line of business contains a reference implementation that you can use to accelerate your implementation. This line of business includes reference implementations for jobs, policy file screens, sample rating rules, sample eligibility rules, and forms logic. The reference implementation also provides sample content.

Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can build your own compliance system. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. The default application contains a sample set of these classifications.

This topic includes:

- “Commercial Package Screens” on page 204
- “Commercial Package Object Model” on page 212
- “Commercial Package Product Model” on page 213

See also

- “Commercial Property” on page 215
- “General Liability” on page 231
- “Inland Marine” on page 245

Commercial Package Screens

The commercial package policy implementation contains a series of screens to create a commercial package policy containing general liability, commercial property, and inland marine lines of business. This section provides descriptions of fields that define the policy contract and contain related information in the default configuration.

This section describes the following screens in commercial package policy:

- “Offerings Screen for Commercial Package” on page 204
- “Qualification Screen for Commercial Package” on page 204
- “Policy Info Screen for Commercial Package” on page 205
- “Line Selection Screen for Commercial Package” on page 206
- “Locations Screen for Commercial Package” on page 207
- “Line of Business Screens for Commercial Package” on page 207
- “Modifiers Screen for Commercial Package” on page 208
- “Risk Analysis Screen for Commercial Package” on page 208
- “Quote Screen for Commercial Package” on page 209
- “Forms Screen for Commercial Package” on page 210
- “Payment Screen for Commercial Package” on page 210

Offerings Screen for Commercial Package

The **Offerings** screen contains a set of questions related to offerings.

Offerings let you define different product types for different types of buyers. Answers to the questions can affect which offerings are available. Offerings can filter parts of the product model such as policy terms, policy lines in a package policy, coverages, covterms, covterm options and packages, modifiers, and question sets.

In the default implementation, the answers to the questions determine the choices on the **Offering Selection** drop-down list.

For more information about offerings, see “Understanding Offerings” on page 477.

In commercial package policy, you must select an offering. Answering all questions in the least risky way results in the most offerings being available. Commercial package policy contains the following offerings:

- Highly Protected Risk
- Premium Program
- Special Risk
- Standard

Qualification Screen for Commercial Package

This screen contains a set of questions to pre-qualify the applicant.

The questions reflect risks that the carrier wants to assess at the beginning of the submission. Since this is the gateway to the balance of the submission, the intent is to determine eligibility of the applicant.

The questions do not contain any regulatory requirements. In your implementation, the question set can impact later functionality and contain regulatory requirements.

In the default implementation, some of the pre-qualification questions raise underwriting issues if the answer is not the correct answer. In Studio you can specify the correct answer and whether or not to raise an underwriting issue.

Policy Info Screen for Commercial Package

The **Policy Info** screen captures basic information about the policy.

This screen contains information about the primary named insured, producer information, policy details, and selected underwriting company. The buttons and choices that you see on the **Policy Info** screen vary based on the contacts associated with the account and how they are associated with the policy. For example, under **Primary Named Insured** you might see any or all of the following:

- New Company
- New Person
- From Address Book
- Existing Contact

Not all choices appear at any given time.

The following table describes key fields of the **Policy Info** screen.

Name of field	Description
Primary Named Insured	<p>PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select:</p> <ul style="list-style-type: none">• New Company• New Person• From Address Book• Existing Contact <p>If the named insured is a person, the social security number is required in Official IDs. If the named insured is a business, then FEIN is required.</p> <p>For Existing Contact, PolicyCenter lists contacts with the Account Holder or Named Insured role on the account. The list does not include contacts who are already the primary or additional named insureds on this policy.</p>
Organization Type	Enables you to select the type of organization such as whether this business is a corporation or partnership.
Additional Named Insureds	Use to create additional named insureds. This might be a business partner, for example. The Add button enables you to select: <ul style="list-style-type: none">• New Company• New Person• From Address Book• Existing Additional Named Insured• Other Contacts
Policy Details	Includes the Term Type , Effective Date , Expiration Date , Written Date , and the Base State .
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p> <p>If you change the producer in the middle of a policy period, the new producer is the Producer of Service. The original Producer of Record remains. When the policy is renewed, the Producer of Service becomes the Producer of Record.</p>
Underwriting Companies	The underwriting company is automatically set for a policy version based on logic defined in segmentation classes. You can select a different underwriting company if you have the correct permissions. <p>The drop-down list contains a configurable set of choices.</p>
Preferred Currency	If PolicyCenter is configured as a multicurrency system, the Policy Info screen displays this label and the following fields related to the preferred currency.

Name of field	Description
Coverage	The preferred or default currency for coverages on the policy. The Coverage currency choices come from the policy line configuration in Product Designer. In the base configuration, the default is the preferred coverage currency on the account, Account.PreferredCoverageCurrency.
Settlement	The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the Quote and other screens. In the base configuration you can select one of the following currencies: <ul style="list-style-type: none"> • USD • EUR • GBP • CAD • AUD • RUB • JPY The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, Account.PreferredSettlementCurrency. You can set the preferred settlement currency in the Account File Summary screen Currencies → Settlement field.

Line Selection Screen for Commercial Package

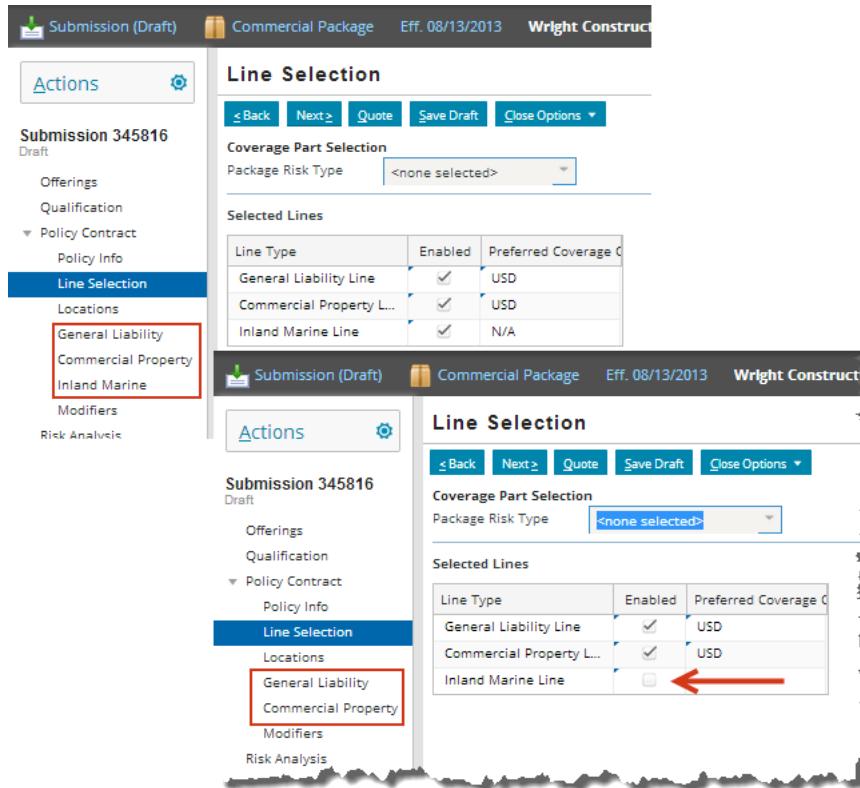
The **Line Selection** screen allows you to enable and disable lines, and to select the risk type.

The **Coverage Part Selection** allows you to select a **Package Risk Type**. The choices are:

- Apartment
- Contractor
- Industrial/Processing
- Institutional
- Mercantile
- Motel/Hotel
- Office
- Services

The **Selected Lines** section displays each line in commercial package policy. PolicyCenter enables the line based on the selected offering. In the default configuration, all offerings enable all lines.

You can remove a line from a policy by clearing the **Enabled** check box. A pop-up window appears asking you to confirm your intention. If you select **OK**, the line is removed from the policy and no longer appears in the left sidebar. Any additions or selections you made in the line are immediately lost.



Locations Screen for Commercial Package

In the **Locations** screen, enter the locations that this policy covers.

By default, the primary location defaults to the primary location on the account. Add other locations that are covered by this policy. If you create a new location, PolicyCenter adds it to the account. Select a location and click **Set As Primary** to change the primary location.

The locations shown on this screen are defined as locations for the policy as a whole, but not necessarily utilized by each line. The commercial property and inland marine lines require additional data related to a location.

All locations added in the **Locations** screen can be added to the commercial property and inland marine lines from the **Buildings and Locations** screen. Select **Add Location** → **Existing Location** to select a policy-wide location.

Locations added directly to the **Buildings and Locations** screen for the commercial property or inland marine lines are automatically included in the **Locations** screen for the commercial package policy.

If the general liability line is enabled, locations have an additional **General Liability Line Territory Code** field. If the commercial property line is enabled, locations have an additional **Commercial Property Line Territory Code** field.

See also

- “Locations” on page 343

Line of Business Screens for Commercial Package

The left sidebar displays each line of business that you selected in the **Line Selection** screen. In the illustration below, the purple box shows each line of business. You can click a line of business to reveal the names of screens

specific to that line. PolicyCenter displays the names for only one line at a time. As you move through the wizard, the line of business in the left sidebar expands to reveal the current step. In the illustration below, the green box shows the expanded inland marine line. The other lines of business collapse.

In commercial package policy, the **Line Review** screen is the alternate name for the **Policy Review** screen. (**Policy Review** is the name of the screen in policies for individual lines of business.) Each **Line Review** screen displays summary information for one line at a time. In a policy change, the **Policy Review** screen is shown and used to resolve conflicts related to out of sequence changes.



See also

- “General Liability Screens” on page 232
- “Commercial Property Screens” on page 216
- “Inland Marine Screens” on page 246

Modifiers Screen for Commercial Package

The **Modifiers** screen displays modifiers defined at the product level. (General liability and commercial property also have separate screens to display modifiers that are specific to each line.) The commercial package product has one modifier for an individual risk premium modification. The rate factors for this modifier allow you to adjust the rating based on:

- Management
- Location
- Building features
- Premises and equipment
- Employees
- Protection

Risk Analysis Screen for Commercial Package

The **Risk Analysis** screen allows you to enter data that is used to evaluate the risk of the applicant.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- UW Referral Reasons** – This tab appears when viewing a policy. It does not appear in a work order such as a submission. This tab allows you to add an underwriting referral reason to a policy. Underwriting referral reasons are usually only used when a notable condition arises outside of a job, possibly outside of the data that PolicyCenter maintains on the policy. For more information about this tab, see “Working with Underwriting Referral Reasons” on page 416.
- UW Issues** – This tab displays underwriting issues that were raised during the job. For more information on this tab, see “Working with Underwriting Issues” on page 414.

- **Prior Policies** – This tab displays information about prior policies usually with another carrier. You can add information about prior policies.
- **Claims** – This tab allows you to search for loss claims on a related policy or by loss date. When integrated with a claim system, this tab displays claims from the claim system. For more information, see “Viewing Loss Claims for Policies” on page 716.
- **Prior Losses** – This tab displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

Quote Screen for Commercial Package

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen.

This screen is similar to the quote screen for other lines of business. It contains the basic information at the top including the total premium and taxes and surcharges. The bottom of the screen displays a breakdown of the premium.

Within the **Quote** screen, your choices at this point include:

- Editing the policy (and thereby invalidating the current quote)
- Creating a new version of the policy
- Saving a draft of the policy
- Binding or issuing the policy
- Withdrawing, declining, or not taking the policy
- Printing the policy

PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

In addition, Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

If you have Guidewire Rating Management, the **Policy Premium** tab has a **Show Rate Worksheet** button. For more information about this button, see “Rating Worksheets” on page 548 in the *Application Guide*.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Policy Premium Tab

The **Policy Premium** tab displays the premium for each line of business in the policy.

The **Override Rating** button allows you to manually override the premium that the rating engine automatically generates for a policy. This option is only available for inland marine, which is the only line within commercial package that has rating override in the default configuration. For more information, see “Rating Overrides” on page 445.

The **Compact View** and **Extended View** buttons display the rating information in compact or extended form. If the approximate page length is greater than 50, then PolicyCenter displays the compact view. Otherwise, PolicyCenter displays the extended view. The compact view shows summary information on the initial screen and allows the user to click to view details. The extended view displays the details on the initial screen.

Gosu code in the PCF file determines which view to display initially. The panel set that contains the **Compact View** and **Extended View** buttons is `RatingPanelSet.CommercialPackage.IMLine.pcf`.

This PCF file includes a modal panel set. The `initialValue` of the `pageLength` variable determines whether the modal panel set displays initially in drill-down or scroll mode. At top of this screen, click the panel set for this

file to display the **Variables** and **Code** tabs at the bottom of the screen. On the **Variables** tab, the `pageLength` variable calculates an `initialValue`. The **Code** tab sets the view mode based on the value of the `pageLength` variable.

Forms Screen for Commercial Package

The **Forms** screen allows you to associate forms with a policy.

PolicyCenter does not store the content of any forms associated with a policy. Therefore, the **Forms** screen only identifies that zero, one, or more, forms have been associated with the policy. The **Forms** screen simply lists form instances that indicate the physical forms, which are usually printed by an issuance system. Each form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form. The form content is not stored within PolicyCenter.

When you click **Quote** or **Bind**, PolicyCenter infers which forms the policy needs. In the default application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy Forms” on page 461
- “Policy Form Pattern Administration” on page 683 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment Screen for Commercial Package

The **Payment** screen displays payment information for the policy.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- **Total Premium**
- **Taxes and fees**
- **Total Cost** – The sum of the previous fields.

Billing

Under the **Billing** heading, you can select:

- **Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- **Alt Billing Account** – Select an alternate billing account.
- **Billing Contact** – Select a person or company as the billing contact.

Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- **List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer's mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample billing accounts. If you are integrated with **BillingCenter**, the plugin returns a list of billing accounts retrieved from **BillingCenter**.

- **Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent's account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle's account as the alternate billing account. The uncle's account is not a subaccount of the parent's account.

You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.

- **Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample installments plans. If you are integrated with **BillingCenter**, the plugin returns a list of valid installments plans for the selected list bill payer. The integration retrieves installments plans from **BillingCenter**.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the `StandAloneBillingSystemPlugin` or the default integration with BillingCenter, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The `StandAloneBillingSystemPlugin` simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payments system, “Payment Integration” on page 553 in the *Integration Guide*.

If you select **Automatic** for **Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.
- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

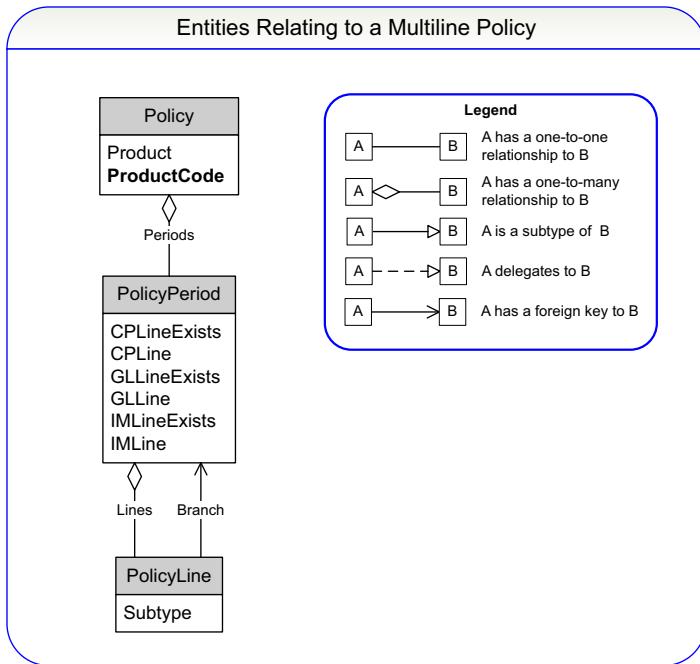
Note: The BillingCenter integration does not support this functionality.

Commercial Package Object Model

This section describes the objects or entities associated with the commercial package line of business.

PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are

easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.



The **Policy** entity has a **Product** field which returns the product associated with the policy. The **ProductCode** fields for a commercial package policy returns **CommercialPackage**. This is the **Code** defined in **Commercial Package** located in **Product Model → Products** in Product Designer.

The **PolicyPeriod** entity has an *LNLlineExists*, where *LNL* is an abbreviation for the line. This derived property returns true if a line exists on the policy period. The *LNLline* derived property allows you to retrieve the policy line. The **PolicyPeriod** entity also has a **Lines** array to each **PolicyLine**.

See also

- “Core Entities Associated with Policies” on page 309
- “Commercial Property Object Model” on page 225
- “General Liability Object Model” on page 242
- “Inland Marine Object Model” on page 254

Commercial Package Product Model

Various features of the commercial package policy are defined in the product model.

The product model is the PolicyCenter feature that identifies the different types of policies, or products, that a given instance of PolicyCenter offers. For each product, the product model details all of the choices around what can be covered. You can think of the product model as a product configuration.

In Studio, the following files in the **Product Model** folder are related to commercial package policy:

Folder	File
Products	Commercial Package
Policy Lines	Commercial Property Line
	General Liability Line
	Inland Marine
Question Sets	CPP Offering
	GL PreQualification
	IM Contractors Equipment Questions

The commercial package policy is a multi-line product that is defined in the product model in Product Designer. Navigate to **Product Model** → **Products** and open **Commercial Package**.

On the **Commercial Package** screen, **Offering Required** is set to true. This setting means that you must select an offering on the **Offerings** screen. Four offerings are defined on the **Offerings** tab.

In **Policy Lines**, the policy lines for this product are specified. The policy lines in this product are **Commercial Property Line**, **General Liability Line**, and **Inland Marine Line**.

In **Question Sets**, questions sets for each line are included. Also included are offering questions that are related to the commercial package product and are not specific to any particular policy line.

On the **Modifiers** screen, you can define product level modifiers. You can specify rate factors which can increase or decrease the premium for the policy.

Commercial Package Forms

There are policy forms associated with the commercial package product. In PolicyCenter, navigate to **Administration** → **Policy Forms**. For more information about product forms, see “**Policy Forms**” on page 461.

Commercial Property

The commercial property line of business is part of the PolicyCenter base configuration.

The commercial property line of business provides coverage against loss or loss of use of buildings and related items (such as contents) because of fire, storms, theft, and other events. Commercial property is similar to the property coverage portion of the businessowners line of business. To accommodate all types of businesses, commercial property offers more types of coverages.

Typically, a single policy covers businesses operating at multiple locations. However, carriers vary in whether they require separate policies for locations that serve different functions and have different risk profiles.

This line of business contains a reference implementation that you can use to accelerate your implementation. This line of business includes reference implementations for jobs, policy file screens, sample rating rules, sample eligibility rules, and forms logic. The reference implementation also provides sample content.

Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can build your own compliance system. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. The default application contains a sample set of these classifications.

This topic includes:

- “Working with Commercial Property” on page 215
- “Commercial Property Screens” on page 216
- “Commercial Property Object Model” on page 225
- “Configuring Commercial Property” on page 229

Working with Commercial Property

This topic provides step-by-step instructions on how to work with a commercial property policy in the user interface.

Adding Locations and Buildings

You can add new or existing buildings and locations. You must specify at least one building for each location.

To add a location

1. In a commercial property policy, navigate to the **Buildings and Locations** screen.

2. Select **Add Location** and choose **New Location** or **Existing Location**.

If you removed a location, you can add back a location by selecting **Add Location** → **Existing Location**.

When adding a location, if the list of existing locations is more than 10, a **More Locations** menu item appears after the tenth location.

To add a building

1. In the **Actions** column for a location, click the control and select **Add Building**. Select **New Building** or **Existing Building**.

If you remove a building, you can add that building back by selecting **Add Building** → **Existing Building**.

When adding a building, if the list of existing buildings is more than 10, a **More Buildings** menu item appears after the tenth building. If you select this menu item, PolicyCenter displays a **More Buildings Selection** screen that displays the buildings.

Removing Buildings

1. In a commercial property policy, navigate to the **Buildings and Locations** screen.

2. Click to add a check mark to the buildings you want to remove.

3. Click **Remove Buildings**.

Copying Coverages to Other Buildings

You can copy coverages from one building to other buildings in the policy. This copies the full coverage pattern on the building. For example, if a coverage is selected on the “copy from” building, that coverage is copied to all “copy to” buildings. In addition, if a coverage is not selected on the “copy from” building, that lack of selection is copied to all the “copy to” buildings.

Product model synchronization removes coverages that are not available in the destination risk.

1. In a commercial property policy, navigate to the **Buildings and Locations** screen.

2. Select **Copy Coverages**. This button is available if there are at least two buildings on the policy.

3. On the **Copy Coverages** screen, use the **Choose Building** drop-down menu to select the building to copy from.

4. Select the buildings to copy to or select **Copy To All**.

See also

- “Configuring Copying of Coverages for Commercial Property” on page 229

Commercial Property Screens

The commercial property implementation contains a series of screens to describe locations and buildings, choose coverages, assess risk, quote the policy, and select payment options. This section provides descriptions of fields that define the policy contract and contain related information in the default configuration.

This section describes the following screens in commercial property.

- “Policy Info Screen for Commercial Property” on page 217
- “Buildings and Locations Screen for Commercial Property” on page 218
- “Blankets Screen for Commercial Property” on page 219
- “Modifiers Screen for Commercial Property” on page 220
- “Risk Analysis Screen for Commercial Property” on page 220
- “Policy Review Screen for Commercial Property” on page 221
- “Quote Screen for Commercial Property” on page 221
- “Forms Screen for Commercial Property” on page 222
- “Payment Screen for Commercial Property” on page 222

Policy Info Screen for Commercial Property

The **Policy Info** screen captures basic information about the policy.

This screen contains information about the primary named insured, producer information, policy details, and selected underwriting company. The buttons and choices that you see on the **Policy Info** screen vary based on the contacts associated with the account and how they are associated with the policy. For example, under **Primary Named Insured** you might see any or all of the following:

- New Company
- New Person
- From Address Book
- Existing Contact

Not all choices appear at any given time.

The following table describes key fields of the **Policy Info** screen.

Name of field	Description
Primary Named Insured	<p>PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select:</p> <ul style="list-style-type: none">• New Company• New Person• From Address Book• Existing Contact <p>If the named insured is a person, the social security number is required in Official IDs. If the named insured is a business, then FEIN is required.</p> <p>For Existing Contact, PolicyCenter lists contacts with the Account Holder or Named Insured role on the account. The list does not include contacts who are already the primary or additional named insureds on this policy.</p>
Organization Type	Enables you to select the type of organization such as whether this business is a corporation or partnership.
Additional Named Insureds	Use to create additional named insureds. This might be a business partner, for example. The Add button enables you to select: <ul style="list-style-type: none">• New Company• New Person• From Address Book• Existing Additional Named Insured• Other Contacts
Policy Details	Includes the Term Type , Effective Date , Expiration Date , Written Date , and the Base State .

Name of field	Description
Producer of Record	The Organization defaults to the producer that you selected on the New Submissions screen. Although you can change the values for Organization and Producer Code , PolicyCenter limits your choices by user permissions. If you change the producer in the middle of a policy period, the new producer is the Producer of Service . The original Producer of Record remains. When the policy is renewed, the Producer of Service becomes the Producer of Record .
Underwriting Companies	The underwriting company is automatically set for a policy version based on logic defined in segmentation classes. You can select a different underwriting company if you have the correct permissions. The drop-down list contains a configurable set of choices.
Preferred Currency	If PolicyCenter is configured as a multicurrency system, the Policy Info screen displays this label and the following fields related to the preferred currency.
Coverage	The preferred or default currency for coverages on the policy. The Coverage currency choices come from the policy line configuration in Product Designer. In the base configuration, the default is the preferred coverage currency on the account, <code>Account.PreferredCoverageCurrency</code> .
Settlement	The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the Quote and other screens. In the base configuration you can select one of the following currencies: <ul style="list-style-type: none"> • USD • EUR • GBP • CAD • AUD • RUB • JPY The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, <code>Account.PreferredSettlementCurrency</code> . You can set the preferred settlement currency in the Account File Summary screen Currencies → Settlement field.

Buildings and Locations Screen for Commercial Property

The **Buildings and Locations** screen allows you to add locations and buildings to the policy. By default, PolicyCenter adds the primary account location. (If the primary account location is not associated with the policy, you can remove it from the policy.)

This screen has three buttons at the top:

- **Add Location** – For step-by-step instructions, see “Adding Locations and Buildings” on page 216.
- **Remove Buildings** – For step-by-step instructions, see “Removing Buildings” on page 216.
- **Copy Coverages** – For step-by-step instructions, see “Copying Coverages to Other Buildings” on page 216.
- **Spreadsheet** – Import or export buildings and locations from a policy in PolicyCenter to a spreadsheet in `.xlsx` format. For step-by-step-instructions, see “Using Spreadsheets Generated by Policy Data Spreadsheet Import/Export” on page 465.

For each building, the **Building and Locations** screen displays the limit for the following coverages:

- **Building Coverage**
- **Business and Personal Property Coverage**
- **Business Income Coverage**
- **Extra Expense Coverage**

The screen does not display the limit for the **Business and Personal Property - Separation of Coverage (Stock)** coverage.

The **Buildings and Locations** and **Building** screens have a **Coverages in** drop-down list.

When multicurrency display is enabled, you can set the currency on a coverable in the user interface.

PolicyCenter then copies this coverage currency to each coverage on the coverable. The main screen for each coverable has the following multicurrency field:

Field	Description
Coverages in	Select the currency for the coverages on this coverable. By default, this value is set to the Policy Info → Preferred Currency → Coverage field. The drop-down list contains the Available Coverage Currencies defined in Product Designer on the main screen of the policy line.

See also

- “Multicurrency Policies” on page 517

Building Screen

The building screen has **Details**, **Coverages**, and **Additional Interest** tabs.

The **Details** tab includes fields for the class code, coverage form, rating, construction details, and improvements.

The **Coverage Form** menu item allows you to select:

- Building and Personal Property**
- Condominium Association**
- Condominium Unit-Owners** – If you select this coverage form, certain coverages cannot be added to the policy. If these coverages are currently on the policy, they are removed. These coverages are:
 - Building Coverage**
 - Business and Personal Property Coverage**
 - Business Personal Property - Separation of Coverage (Stock)**

The **Coverages** tab allows you to add or remove coverages, and specify coverage details such as the limit.

The **Additional Interest** tab allows you to add or remove additional interests.

Blankets Screen for Commercial Property

Blanket insurance provides coverage for a combination of items with a single limit for all the covered items that are included in the blanket. The **Blankets** screen displays blankets applied to the policy. Blankets are grouped into two types: direct loss and time element. Each blanket contains multiple coverages. The coverages in one type of blanket cannot be combined with the coverages in the other. You can apply the blanket to one or more locations. You can add a blanket that has a single coverage.

Blanket Details Screen

Click **Add Blanket** to display the **Blanket Details** screen. PolicyCenter provides an autonumber for the blanket. You can also enter a **Description**.

In the **Blanket Type** drop-down menu, you can apply the blanket coverage in the following ways:

- Single Location** – Applies selected coverages to a single location which you specify in the **Location** field.
- Multiple Locations** – Applies selected coverages to multiple locations and buildings.
- Single Coverage** – Applies the same coverage across multiple locations and buildings. For example, you can select a single coverage of only **Building Coverage**, or only **Business Personal Property Coverage**. Select **Direct Loss** or

Time Element from the Group Type drop-down menu. Then select a coverage from the Building Coverage drop-down menu.

You define the Group Type choices in the product model in Studio. The drop-down menu allows you to select two types of blankets:

- Direct Loss – Provides coverages for buildings. The direct loss coverages are:
 - Building Coverage
 - Business Personal Property Coverage
 - Business Personal Property - Separation of Coverage (Stock)
- Time Element – Provides the following coverages:
 - Business Income Coverage
 - Extra Expense Coverage

You can also specify a Limit, Deductible, and Coinsurance percentage.

Click Show Coverages to display the coverages in the blanket. You can include or remove a coverage from the blanket by selecting the coverage and clicking Include Selected in Blanket or remove Selected from Blanket. All blankets must have at least two coverages.

Configuring Blankets in the Product Model

You can configure blankets in Product Designer. Navigate to **Product Model** → **Policy Lines** → **Commercial Property Line**.

The limit, deductible, and coinsurance percentage are configured in Studio. On the **Coverages** screen, the **CPBlanket Coverage** defines these values.

For each coverage, you can specify a blanket group type. On the **Coverages** screen, select a coverage. On the coverage, select **Direct Loss** or **Time Element** for **Blanket Group Type**. If a value is not selected, then you cannot add this coverage to a blanket.

Modifiers Screen for Commercial Property

The commercial property line has one modifier for schedule credits. The rate factors for this modifier allow you to adjust the rating based on:

- Building features
- Employees
- Management
- Premises and equipment
- Protection
- Risk elements not addressed in the classification plan

Risk Analysis Screen for Commercial Property

The Risk Analysis screen allows you to enter data that is used to evaluate the risk of the applicant.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – This tab appears when viewing a policy. It does not appear in a work order such as a submission. This tab allows you to add an underwriting referral reason to a policy. Underwriting referral reasons are usually only used when a notable condition arises outside of a job, possibly outside of the data that PolicyCenter maintains on the policy. For more information about this tab, see “Working with Underwriting Referral Reasons” on page 416.
- **UW Issues** – This tab displays underwriting issues that were raised during the job. For more information on this tab, see “Working with Underwriting Issues” on page 414.
- **Prior Policies** – This tab displays information about prior policies usually with another carrier. You can add information about prior policies.

- **Claims** – This tab allows you to search for loss claims on a related policy or by loss date. When integrated with a claim system, this tab displays claims from the claim system. For more information, see “Viewing Loss Claims for Policies” on page 716.
- **Prior Losses** – This tab displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

The default configuration does not contain underwriting issues for commercial property. Therefore, the **UW Issues** screen will not have any issues unless you add them to the configuration.

Policy Review Screen for Commercial Property

The Policy Review screen is similar to the summary screen for other lines of business.

The Policy Review screen displays general information about the policy including buildings and locations. This is the same information that appeared on the **Buildings and Locations** screen. Each building has a link to the **Details**, **Coverages**, and **Additional Interest** tabs.

For a submission job, this screen contains all the policy data in summary form. For other jobs, this screen displays the differences between the policy versions. It is useful to review this screen before generating a quote. If any of the information needs to be changed, click the menu link on the left sidebar to edit the appropriate screen.

If PolicyCenter is configured as a multicurrency system, the Policy Review screen displays the cost for each coverable’s coverages in the currency set on the coverable.

Quote Screen for Commercial Property

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen.

This screen is similar to the quote screen for other lines of business. It contains the basic information at the top including the total premium and taxes and surcharges. The bottom of the screen displays a breakdown of the premium.

Within the **Quote** screen, your choices at this point include:

- Editing the policy (and thereby invalidating the current quote)
- Creating a new version of the policy
- Saving a draft of the policy
- Binding or issuing the policy
- Withdrawing, declining, or not taking the policy
- Printing the policy

PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

In addition, Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

If you have Guidewire Rating Management, the **Policy Premium** tab has a **Show Rate Worksheet** button. For more information about this button, see “Rating Worksheets” on page 548 in the *Application Guide*.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

In the default configuration, the rating system rates only the building coverage and the business personal property coverage. Other coverages are not reflected in the price of the policy.

Two to four costs are generated for each selected coverage within a period of time. There may be multiple versions of the cost over time, but none of them overlap. On the **Coverages** tab of the **Building** screen, you can specify the **Cause of Loss** to be **Basic**, **Broad**, or **Special**. The cause of loss determines the costs for the coverage as follows:

- Basic results in two costs, using Group I and Group II rates.
- Broad results in three costs, using Group I, Group II and Broad rates.
- Special results in four costs, using Group I, Group II, Broad, and Special rates.

From the **Quote** page, click on the building link to view the **Cost Details** screen for each building.

The standard rates in the default configuration are as follows:

Coverage	Group I rate	Group II rate	Broad	Special
Building Coverage	0.12	0.08	0.08	0.08
Business Personal Property Coverage	0.15	0.15	0.08	0.08

Policy Premium Tab

The **Override Rating** button allows you to manually override the premium that the rating engine automatically generates for a policy. For more information, see “Rating Overrides” on page 445.

Forms Screen for Commercial Property

The **Forms** screen allows you to associate forms with a policy.

PolicyCenter does not store the content of any forms associated with a policy. Therefore, the **Forms** screen only identifies that zero, one, or more, forms have been associated with the policy. The **Forms** screen simply lists form instances that indicate the physical forms, which are usually printed by an issuance system. Each form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form. The form content is not stored within PolicyCenter.

When you click **Quote** or **Bind**, PolicyCenter infers which forms the policy needs. In the default application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy Forms” on page 461
- “Policy Form Pattern Administration” on page 683 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment Screen for Commercial Property

The **Payment** screen displays payment information for the policy.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- **Total Premium**
- **Taxes and fees**
- **Total Cost** – The sum of the previous fields.

Billing

Under the **Billing** heading, you can select:

- **Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- **Alt Billing Account** – Select an alternate billing account.
- **Billing Contact** – Select a person or company as the billing contact.

Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- **List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer’s mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the `StandAloneBillingSystemPlugin`, the plugin returns a list of sample billing accounts. If you are integrated with BillingCenter, the plugin returns a list of billing accounts retrieved from BillingCenter.

- **Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent’s account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle’s account as the alternate billing account. The uncle’s account is not a subaccount of the parent’s account.

You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.

- **Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample installments plans. If you are integrated with **BillingCenter**, the plugin returns a list of valid installment plans for the selected list bill payer. The integration retrieves installments plans from **BillingCenter**.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the **StandAloneBillingSystemPlugin** or the default integration with **BillingCenter**, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The **StandAloneBillingSystemPlugin** simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payments system, “Payment Integration” on page 553 in the *Integration Guide*.

If you select **Automatic** for **Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.

- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

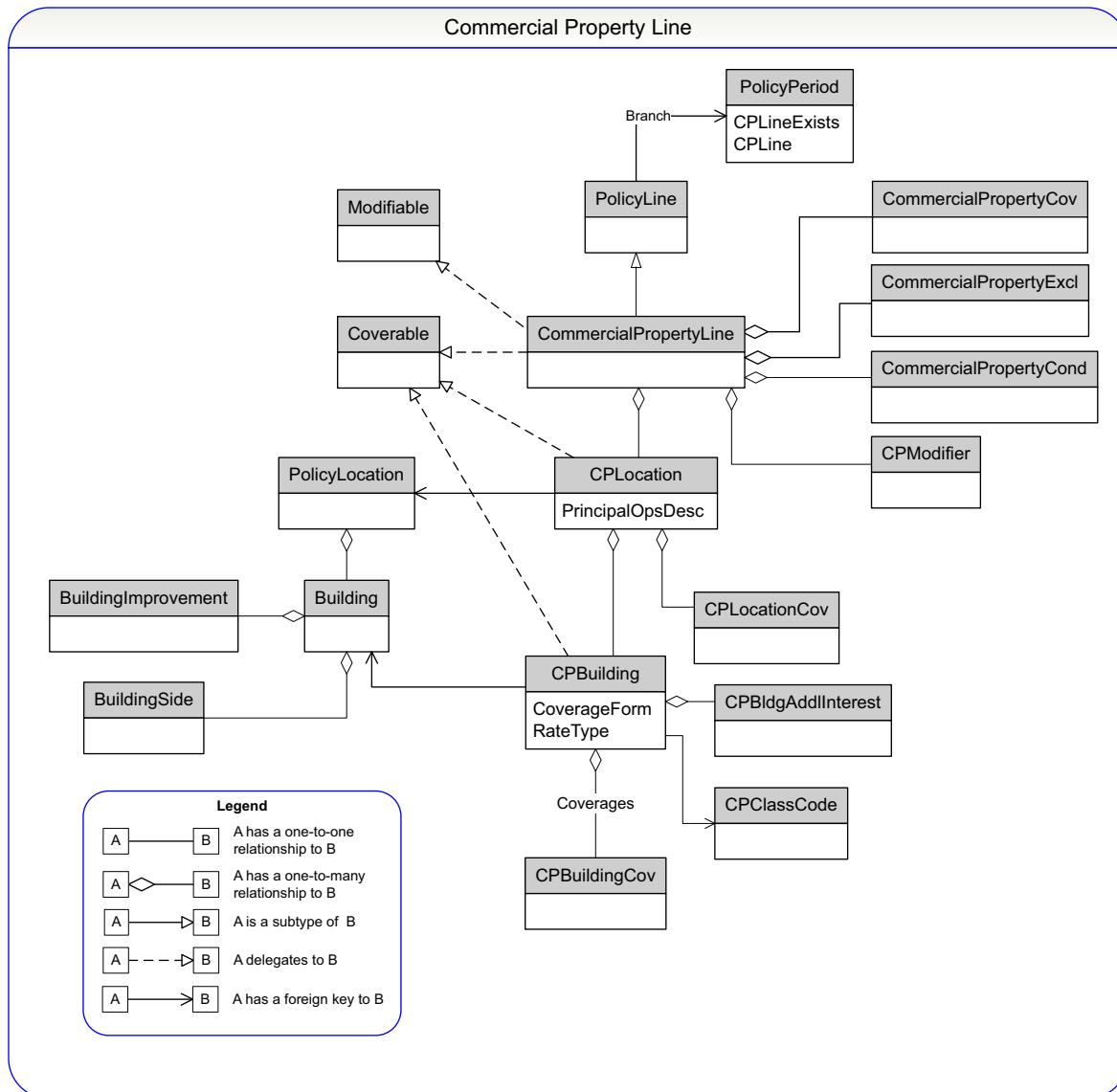
Note: The BillingCenter integration does not support this functionality.

Commercial Property Object Model

This section describes the objects or entities associated with the commercial property line of business.

PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.

The following diagram of the commercial property object model shows how some of the key entities relate to each other.



See also

- “Core Entities Associated with Policies” on page 309
- “Cost and Transaction Model for Commercial Property Line” on page 426

Line Entity in Commercial Property

The `CommercialPropertyLine` entity is an entity subtype of `PolicyLine`. This entity has an array of `CPLocation` entities.

The commercial property line delegates to `Coverable`, so you can add coverages to the line. In the default configuration, there are no line-level coverages defined. However, if you add line-level coverages, the `CommercialPropertyLine` entity has arrays for `CommercialPropertyCov`, `CommercialPropertyExcl`, and `CommercialPropertyCond` that you can use. For more information, see “Coverage Entities in Commercial Property” on page 226.

The commercial property line delegates to `Modifiable`, so you can add modifiers to the line. The `CommercialPropertyLine` entity has an array of `CPModifier` entities. For more information, see “Modifier Entity in Commercial Property” on page 227.

Location Entity in Commercial Property

The `CPLocation` entity identifies a location on the line through a foreign key reference to a `PolicyLocation` entity. The `CPLocation` entity delegates to `Coverable`, so you can add coverages to it. In the default configuration, no location-level coverages are defined. The `PrincipalOpsDesc` field is a text field for describing the principal types of operations and occupancy that occur at this location. The `CPLocation` entity has an array of `CPBuilding` entities.

Building Entity in Commercial Property

The `CPBuilding` entity identifies a building at a location through a foreign key reference to a `Building` entity. The `CPBuilding` entity delegates to `Coverable`, so you can add coverages to it. The default configuration defines a number of building coverage types. The `CPBuilding` entity has an array key to `CPBuildingCov` entities. The `CPBuilding` entity has an array key to `CPBldgAddlInterest` entities.

Coverage Entities in Commercial Property

There are several coverage entities that define the types of coverage terms that can be assigned for the commercial property line. The `CommercialPropertyCov` and `CPLocationCov` entities are for line-level and location-level coverages, respectively, and are placeholders for customization. The `CPBuildingCov` is for building-level coverages. The `CPBuildingCov` entity is for coverages that apply to `CPBuildings`.

The following coverages are provided in the default configuration:

Coverage	Description
Building Coverage	This coverage provides insurance for the building structure itself. This is a suggested coverage that is not available if the coverage form is condominium unit-owners.
Business Income Coverage	This coverage is a suggested coverage that provides protection for business income. This coverage is not available if the coverage form is Condominium Association.

Coverage	Description
Business Personal Property Coverage	<p>This coverage provides insurance for the following types of property that are located in the building or close to the building:</p> <ul style="list-style-type: none"> • Stock • Machinery & Equipment • Fixtures, improvements and alterations • Tenants Betterments and Improvements • Property of Others <p>This is a suggested coverage that is not available if the coverage form is condominium unit-owners.</p>
Business Personal Property - Separation of Coverage (Stock) Coverage	<p>This coverage enables the insured to choose different coverage terms for certain items in their stock. The insured may choose this coverage if they have stock that is of higher value than the rest of their business personal property. For example, a company that sells computer chips will want to insure computer chips at a higher limit than other types of business personal property.</p> <p>This is a suggested coverage that is not available if the coverage form is condominium unit-owners or condominium association.</p>
Extra Expense Coverage	This coverage provides insurance against extra expenses. It is a suggested coverage.

Modifier Entity in Commercial Property

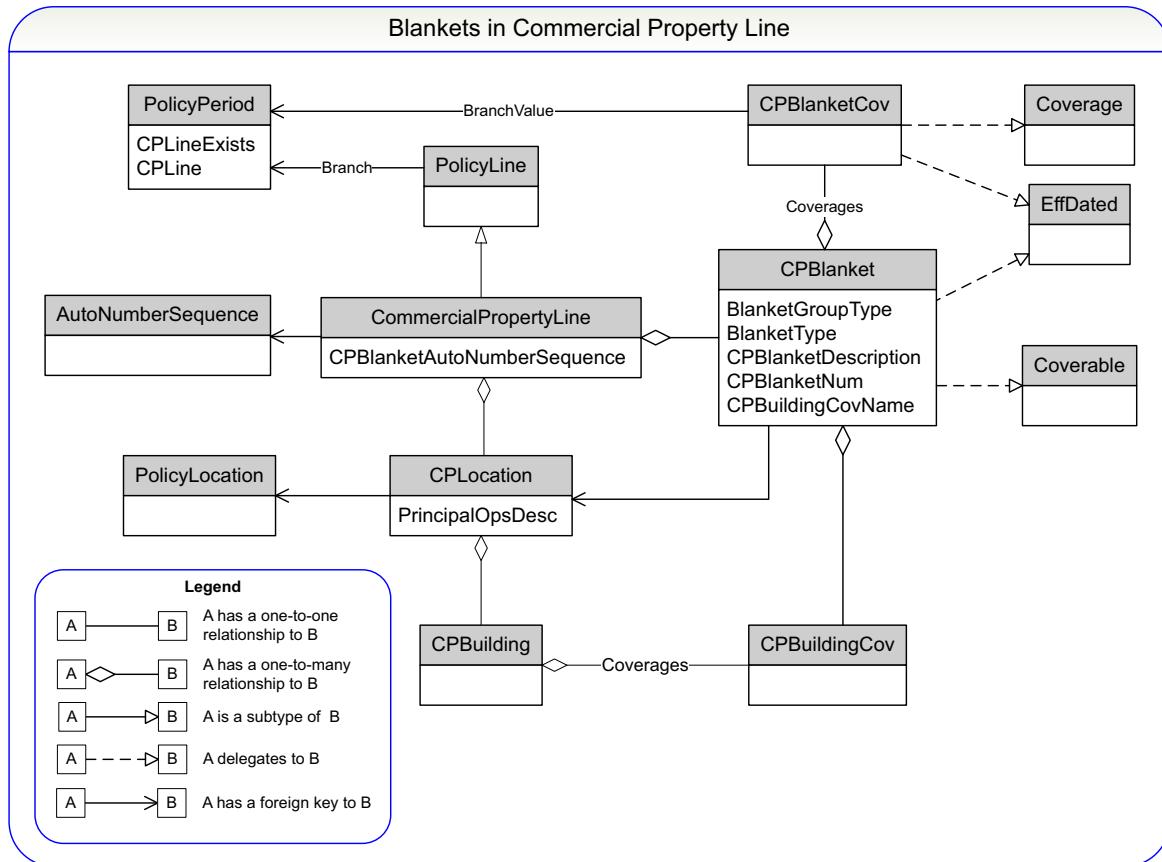
The CPModifier entity represents modifiers on the commercial property policy line. In the default configuration, the commercial property line has a single modifier defined, CPScheduleCredits.

You can view and configure this modifier in Product Designer by navigating to **Product Model → Policy Lines → Commercial Property Line**. Go to the **Modifiers** page, then click the **Schedule Rates** modifier. Go to the **Rate Factors** page. You can define schedule rates for a modifier on this page. Schedule rates define the credits and debits to apply when calculating the quote. For the **CPScheduleCredits** modifier, the total value of the schedule rates must be between -0.25 and 0.25. The following tables lists schedule rates with minimum and maximum values:

Modifier	Value
Building features	-0.05 to 0.05
Employees	-0.03 to 0.03
Management	-0.08 to 0.08
Premises and equipment	-0.05 to 0.05
Protection	-0.02 to 0.02
Risk elements not addressed in classification plan	-0.12 to 0.12

Blanket Entities in Commercial Property

The following illustration shows how the blanket coverage entities relate to each other in the commercial property line. The illustration shows the delegates for the CPBlanket and CPBlanketCov entities. The illustration does not show the delegates for other entities.



The **CommercialPropertyLine** entity has an array key to the **CPBlanket** entity. The **CPBlanketAutoNumberSequence** property is a foreign key to the **AutoNumberSequence** entity. This property is used to auto-number the blankets.

The **CPBlanket** entity is the main entity for blanket coverage. This entity delegates to **EffDated** and **Coverable**. It has array keys to **CPBlanketCov** and **CPBuildingCov**. The **CPBlanket** entity has a foreign key to the **CPLocation** entity. This foreign key is null if there are multiple locations.

The **CPBlanket** entity has three arrays to **CPBuildingCov** entities:

- **BuildingCoverages** – Retrieves coverages that are currently included in a blanket.
- **BuildingCoveragesByBlanketType** – Retrieves all the coverages that can be included on the current blanket. If the blanket covers a single location, this retrieves coverages based on group type and location. If the blanket is a single coverage, this retrieves coverages by coverage type.
- **MatchingBuildingCoverages** – Is the same as **BuildingCoverages** except that it can be different momentarily when the user is changing a blanket. When that occurs, the `removeNonMatchingCoverages` method on **CPBlanketEnhancement** compares the two arrays to determine what to add or remove.

The **CPBlanketCov** entity delegates to **Coverage** and **EffDated**.

Configuring Commercial Property

This topic describes how to configure commercial property line of business.

Configuring Copying of Coverages for Commercial Property

When you select to copy coverages from a building, PolicyCenter calls the `copyCoverages` method in `gw.lob.cp.CPBuildingEnhancement`. This method uses the `gw.coverage.AllCoverageCopier` to copy the coverages.

General Liability

The general liability line of business is part of the PolicyCenter base configuration. General liability covers a policyholder for broad categories of liability for third party losses.

This line of business contains a reference implementation that you can use to accelerate your implementation. This line of business includes reference implementations for jobs, policy file screens, sample rating rules, sample eligibility rules, and forms logic. The reference implementation also provides sample content.

Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can build your own compliance system. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. The default application contains a sample set of these classifications.

This topic includes:

- “General Liability Overview” on page 231
- “General Liability Screens” on page 232
- “General Liability Object Model” on page 242

General Liability Overview

The general liability line covers a policyholder for broad categories of liability for third party losses. Final audit is provided in the general liability line.

As is typical of a liability policy, all coverages are line-level coverages. At a minimum, each general liability policy contains basic general liability coverages. In addition, the policy can contain additional coverages, exclusions, conditions, and additional insureds. Additional coverages provide insurance for specific types of liability such as pollution or electronic data. Exclusions allow you to exclude certain types of liability such as damage to rented premises. Conditions allow you to define other contractual obligations on the policy. You can extend the liability coverage by adding coverage for additional insureds. Each additional insured must have a type, such as controlling interest or lessor of leased equipment.

Exposures allow you to quantify the risk at a specific location. You quantify the risk by entering class codes and a basis amount, which is typically annual sales. These basis amounts are audited if the policy requires final audit.

General Liability Screens

The general liability implementation contains a series of screens which allow you to choose coverages, assess risk, quote, and select payment options for a policy. This section provides descriptions of fields that define the policy contract and contain related information.

This section describes the following screens in general liability:

- “Qualification Screen for General Liability” on page 232
- “Policy Info Screen for General Liability” on page 232
- “Locations Screen for General Liability” on page 234
- “Coverages Screen for General Liability” on page 234
- “Exposures Screen for General Liability” on page 237
- “Modifiers Screen for General Liability” on page 238
- “Risk Analysis Screen for General Liability” on page 238
- “Quote Screen for General Liability” on page 239
- “Forms Screen for General Liability” on page 239
- “Payment Screen for General Liability” on page 240

Qualification Screen for General Liability

This screen contains a set of questions to pre-qualify the applicant.

The questions reflect risks that the carrier wants to assess at the beginning of the submission. Since this is the gateway to the balance of the submission, the intent is to determine eligibility of the applicant.

The questions do not contain any regulatory requirements. In your implementation, the question set can impact later functionality and contain regulatory requirements.

In the default implementation, some of the pre-qualification questions raise underwriting issues if the answer is not the correct answer. In Studio you can specify the correct answer and whether or not to raise an underwriting issue.

Policy Info Screen for General Liability

The **Policy Info** screen captures basic information about the policy.

This screen contains information about the primary named insured, producer information, policy details, and selected underwriting company. The buttons and choices that you see on the **Policy Info** screen vary based on the contacts associated with the account and how they are associated with the policy. For example, under **Primary Named Insured** you might see any or all of the following:

- New Company
- New Person
- From Address Book
- Existing Contact

Not all choices appear at any given time.

The following table describes key fields of the **Policy Info** screen.

Name of field	Description
Primary Named Insured	<p>PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select:</p> <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact <p>If the named insured is a person, the social security number is required in Official IDs. If the named insured is a business, then FEIN is required.</p> <p>For Existing Contact, PolicyCenter lists contacts with the Account Holder or Named Insured role on the account. The list does not include contacts who are already the primary or additional named insureds on this policy.</p>
Organization Type	Enables you to select the type of organization such as whether this business is a corporation or partnership.
Additional Named Insureds	Use to create additional named insureds. This might be a business partner, for example. The Add button enables you to select:
	<ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Additional Named Insured • Other Contacts
Policy Details	Includes the Term Type , Effective Date , Expiration Date , Written Date , and the Base State .
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p> <p>If you change the producer in the middle of a policy period, the new producer is the Producer of Service. The original Producer of Record remains. When the policy is renewed, the Producer of Service becomes the Producer of Record.</p>
Underwriting Companies	<p>The underwriting company is automatically set for a policy version based on logic defined in segmentation classes. You can select a different underwriting company if you have the correct permissions.</p> <p>The drop-down list contains a configurable set of choices.</p>
Preferred Currency	If PolicyCenter is configured as a multicurrency system, the Policy Info screen displays this label and the following fields related to the preferred currency.

Name of field	Description
Coverage	The preferred or default currency for coverages on the policy. The Coverage currency choices come from the policy line configuration in Product Designer. In the base configuration, the default is the preferred coverage currency on the account, Account.PreferredCoverageCurrency.
Settlement	The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the Quote and other screens. In the base configuration you can select one of the following currencies: <ul style="list-style-type: none"> • USD • EUR • GBP • CAD • AUD • RUB • JPY The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, Account.PreferredSettlementCurrency. You can set the preferred settlement currency in the Account File Summary screen Currencies → Settlement field.

Locations Screen for General Liability

In the Locations screen, enter the locations that this policy covers.

By default, the primary location defaults to the primary location on the account. Add other locations that are covered by this policy. If you create a new location, PolicyCenter adds it to the account. Select a location and click Set As Primary to change the primary location.

In general liability policies, locations have an additional **General Liability Line Territory Code** field that contains the territory code for the location. The territory code can have an associated rating factor.

For more information about locations, see “Locations” on page 343.

Coverages Screen for General Liability

The Coverages screen allows you to add standard and additional liability coverages as well as exclusions, conditions, and additional insureds.

When multicurrency display is enabled, you can set the currency on a coverable in the user interface. PolicyCenter then copies this coverage currency to each coverage on the coverable. The main screen for each coverable has the following multicurrency field:

Field	Description
Coverages in	Select the currency for the coverages on this coverable. By default, this value is set to the Policy Info → Preferred Currency → Coverage field. The drop-down list contains the Available Coverage Currencies defined in Product Designer on the main screen of the policy line.

See also

- “Multicurrency Policies” on page 517

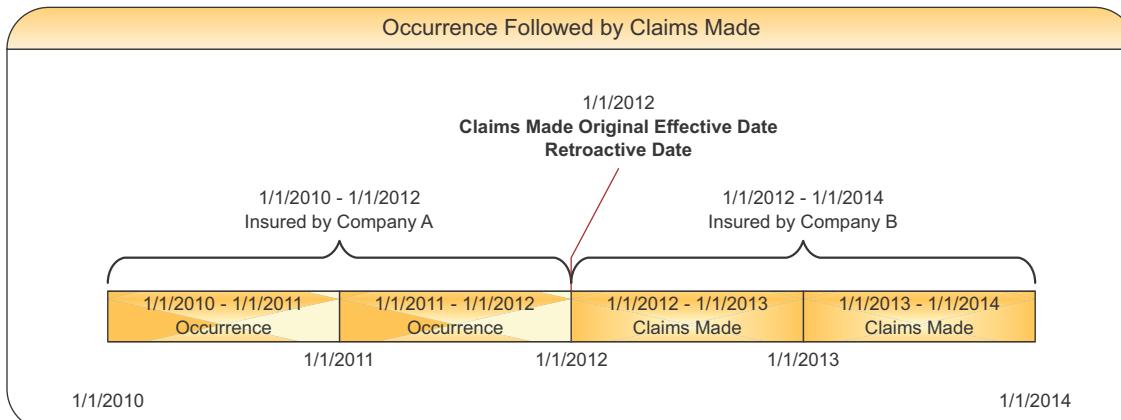
Standard Coverages

The Standard Coverages tab displays the basic general liability coverages that are used by most policies.

Field	Description
Policy Basis	<p>Select either Claims Made or Occurrence.</p> <ul style="list-style-type: none"> If Claims Made is selected, then the policy covers claims filed during the policy term. The claim may be based on a loss which occurred after an explicit retroactive date. The Claims Made Original Effective Date and Retroactive Date fields appear and are required. These fields limit how long before the beginning of the policy period a loss can have occurred and still be eligible for coverage if filed during the period. These fields usually have the same date. <p>The Claims Made Original Effective Date is the date that the policyholder first became covered on a claims made basis. (Prior to that date, they were covered on an occurrence basis or they had no coverage.)</p> <p>Only losses which occurred on or after the Retroactive Date are eligible for coverage. This date would never be earlier than the Claims Made Original Effective Date. Set the Retroactive Date later to limit the period of time for which the carrier provides coverage for claims which occurred in the past.</p> <ul style="list-style-type: none"> If Occurrence is selected, the policy covers losses that occur during the policy term. This type of policy covers a loss that occurs during the policy term, but is reported after the term expires. <p>Note: You cannot change a policy from Claims Made to Occurrence in a mid-term policy change. If you need to make this change, you must cancel the policy, then rewrite it.</p>
Split BI / PD Limits	<p>Yes or No. If No, general liability bodily injury and property damage are covered by a single limit. If Yes, some general liability coverage parts are split into a Bodily Injury (BI) limit and a Property Damage (PD) limit. In the default configuration, the Occurrence Limit, Aggregate Limit, and Product/ Comp.Ops Aggregate are split. For example, if Yes is selected, the Occurrence Limit is split into Bodily Injury Occurrence Limit and Property Damage Occurrence Limit.</p> <p>You can configure which limits are split in Studio. Navigate to Product Model → Policy Lines and click General Liability Line. Click the Basics & Coverages tab. In the left pane, select one of the coverage terms under General Liability. On the Availability tab for the coverage term, the Availability Script determines whether the coverage term is available with split limits.</p>

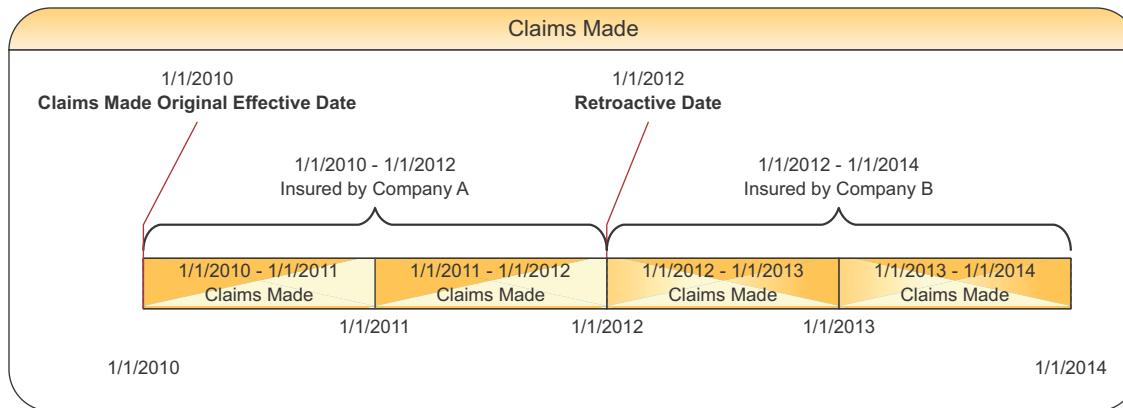
Policy basis examples

The following illustration shows four policy terms of a general liability policy. The insured had an occurrence policy with Company A from 1/1/2010 until 1/1/2012. On 1/1/2012, the insured switched to Company B and changed to a claims made policy. The Claims Made Original Effective Date is set to 1/1/2012, the date that the insured started the claims made policy. The Retroactive Date, the earliest loss date for a claim, is also set to 1/1/2012. From 1/1/2010 to the present, there is no gap in coverage. Losses that occurred prior to the Retroactive Date are covered by Company A, even if the claim is filed after the Retroactive Date. Any losses that occur after the Retroactive Date are covered by Company B.



In the following illustration, the insured had claims made policies with two carriers. The **Claims Made Original Effective** date is set 1/1/2010, the first date that the insured had a claims made policy (with any carrier). The **Retroactive Date** is set to 1/1/2012, the first date the insured has a claims made policy with Company B.

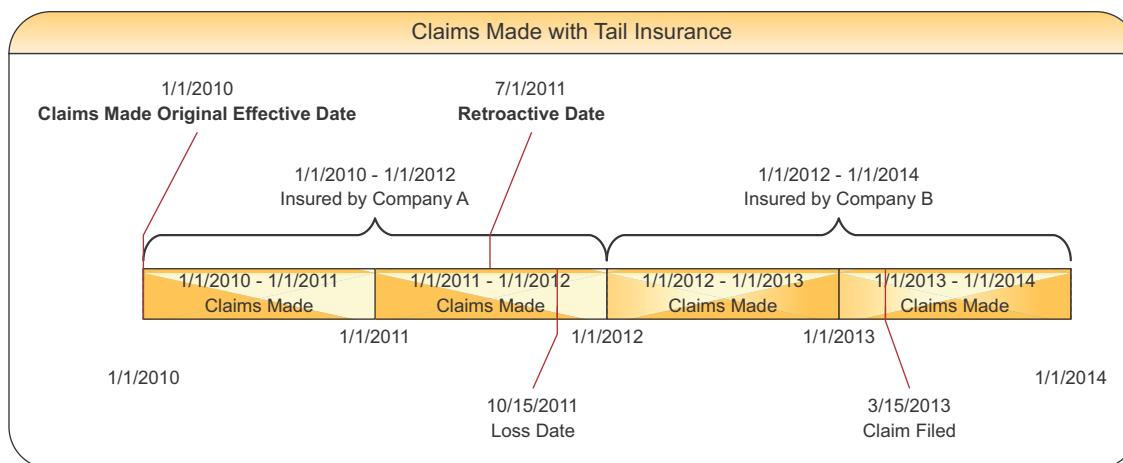
If the Retroactive Date is later than the Claims Made Original Effective Date, then there is a gap in coverage. For example, a loss occurred after the Claims Made Original Effective Date but before the Retroactive Date. The claim is filed after the retroactive date. Company A does not cover this loss because claim is filed after that policy has expired. Company B does not cover this policy because the loss occurred before the retroactive date. Tail insurance can cover this gap.



Tail insurance provides coverage for a gap either by:

- Extending the reporting period for the expired policy. Typically this extension requires the customer to pay an additional premium.
 - Issuing a specific tail coverage policy. This tail coverage policy covers claims that are reported after the last claims made policy has expired.

In the following illustration, the **Retroactive Date** is moved back six months to provide some, but not complete, coverage for the gap in insurance. For example, Company B will cover a claim filed on 3/15/2013 for a loss that occurred on 10/15/2011, prior the date that insurance started with Company B.



To add standard coverages

In PolicyCenter, the **Standard Coverages** tab displays the **General Liability**, **GL Deductible**, and **Arbitration** coverages. You can configure which coverages appear on the **Standard Coverages** tab.

1. In Product Designer, navigate to **Product Model** → **Policy Lines** and open **General Liability Line**.
 2. Go to **Coverages** and click to view a coverage.

3. To define a coverage as standard, set **Category** to **GL REQUIRED**.

Additional Coverages

The **Additional Coverages** tab allows you to select coverages in addition to the coverages on the **Standard Coverages** tab. To define a coverage as additional, set **Category** to a value other than **GL REQUIRED** in Product Designer.

Exclusions and Conditions

This tab allows you to add exclusions and conditions defined for the general liability line of business.

See also

- “Cov erages, Exclusions, Conditions, and Coverables Overview” on page 473

Additional Insureds

The **Additional Insureds** tab allows you to add persons or companies to the general liability policy. The policy extends coverage to those persons or companies based on the additional insured **Type** field. The **Type** field is required. Select and specify a **Type** such as **Lessors**, **Contractors**, or **Vendors**. For example, if you are a manufacturer, you can extend product liability coverage to vendors who sell your product.

Exposures Screen for General Liability

In general liability, exposures provide the basis for rating. Each exposure consists of a policy location, a class code, and a basis value. The exposure fields are described in the following table.

Field	Description
Effective Date	The effective date of the exposure. This value defaults to the effective date of the policy.
Expiration Date	The expiration date of the exposure. This value defaults to the expiration date of the policy.
Location Name	Select a location on the policy.
Class Code	Select a class code.
Description	Description of the selected class code.
Basis	Enter the basis amount.
Basis Type	Basis type for the selected class code.

Splitting or Ending an Exposure in a Policy Change Job

In a policy change, there is sometimes a need to split the basis of an exposure into two rating periods. In other circumstances, you need to end an exposure and replace it with another one. Although this happens infrequently, you can select certain types of exposures and split them or end them at the policy change effective date. PolicyCenter prorates the original basis and displays the prorated value for each rating period. PolicyCenter also prorates the basis when you end an exposure. You can edit the basis amounts.

The basis for an exposure can be rate scalable or basis scalable. In a policy change, you can split or end exposures that are basis scalable.

If an exposure is rate scalable, the length of the policy period does not affect the basis amount. If the length of time is shortened, the rate is adjusted to reflect the shorter period of time, but the basis remains the same. The number of square feet of an office building is an example of an exposure with a rate scalable basis. The square feet remains the same regardless of the length of the policy period.

If an exposure is basis scalable, the length of the policy period can affect the basis amount. Payroll or sales, are examples of exposures that are base scalable. For example, the payroll basis will probably be smaller for a six month period than for a one year period.

Each exposure has a class code and associated basis type. Class codes are specified in **GLClassCode**. You can view class codes in Studio by navigating to **System Tables** and opening `gl_class_code.xml`.

Basis types are specified in **ClassCodeBasis**. You can view the basis types in Studio by navigating to **System Tables** and opening `class_code_basis.xml`. If the **AuditTable** field is **true**, then exposures using class codes with that basis type are basis scalable, and can be split or ended.

To split an exposure in a policy change

The **End** and **Split** buttons appear on the **Exposures** screen in a policy change.

1. Select one or more exposures by adding a check mark in the first column.

2. Click **End** or **Split**.

If you select **Split**, selected exposures that are basis scalable will be split around the effective date of the policy change. The basis is prorated.

If you select **End**, selected exposures that are basis scalable will end on the effective date of the policy change.

Modifiers Screen for General Liability

The **Modifiers** screen allows you to input modifier values.

A modifier is a value used by the rating engine to adjust the policy premium or some portion of the premium. Modifiers capture information relevant to the pricing of a policy that are not necessarily tied to a specific coverable or coverage.

In the default configuration, the general liability line has an experience modifier and a schedule rate modifier.

Risk Analysis Screen for General Liability

The **Risk Analysis** screen allows you to enter data that is used to evaluate the risk of the applicant.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – This tab appears when viewing a policy. It does not appear in a work order such as a submission. This tab allows you to add an underwriting referral reason to a policy. Underwriting referral reasons are usually only used when a notable condition arises outside of a job, possibly outside of the data that PolicyCenter maintains on the policy. For more information about this tab, see “Working with Underwriting Referral Reasons” on page 416.
- **UW Issues** – This tab displays underwriting issues that were raised during the job. For more information on this tab, see “Working with Underwriting Issues” on page 414.
- **Prior Policies** – This tab displays information about prior policies usually with another carrier. You can add information about prior policies.
- **Claims** – This tab allows you to search for loss claims on a related policy or by loss date. When integrated with a claim system, this tab displays claims from the claim system. For more information, see “Viewing Loss Claims for Policies” on page 716.
- **Prior Losses** – This tab displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

Policy Review Screen for General Liability

The **Policy Review** screen is similar to the summary screen for other lines of business.

For a submission job, this screen contains all the policy data in summary form. For other jobs, this screen displays the differences between the policy versions. It is useful to review this screen before generating a quote.

If any of the information needs to be changed, click the menu link on the left sidebar to edit the appropriate screen.

If PolicyCenter is configured as a multicurrency system, the **Policy Review** screen displays the cost for each coverable's coverages in the currency set on the coverable.

Quote Screen for General Liability

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen.

This screen is similar to the quote screen for other lines of business. It contains the basic information at the top including the total premium and taxes and surcharges. The bottom of the screen displays a breakdown of the premium.

Within the **Quote** screen, your choices at this point include:

- Editing the policy (and thereby invalidating the current quote)
- Creating a new version of the policy
- Saving a draft of the policy
- Binding or issuing the policy
- Withdrawing, declining, or not taking the policy
- Printing the policy

PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

In addition, Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

If you have Guidewire Rating Management, the **Policy Premium** tab has a **Show Rate Worksheet** button. For more information about this button, see “Rating Worksheets” on page 548 in the *Application Guide*.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Forms Screen for General Liability

The **Forms** screen allows you to associate forms with a policy.

PolicyCenter does not store the content of any forms associated with a policy. Therefore, the **Forms** screen only identifies that zero, one, or more, forms have been associated with the policy. The **Forms** screen simply lists form instances that indicate the physical forms, which are usually printed by an issuance system. Each form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form. The form content is not stored within PolicyCenter.

When you click **Quote** or **Bind**, PolicyCenter infers which forms the policy needs. In the default application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy Forms” on page 461
- “Policy Form Pattern Administration” on page 683 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment Screen for General Liability

The **Payment** screen displays payment information for the policy.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- Total Premium**
- Taxes and fees**
- Total Cost** – The sum of the previous fields.

Billing

Under the **Billing** heading, you can select:

- Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- Alt Billing Account** – Select an alternate billing account.
- Billing Contact** – Select a person or company as the billing contact.

Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer’s mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the

`StandAloneBillingSystemPlugin`, the plugin returns a list of sample billing accounts. If you are integrated with BillingCenter, the plugin returns a list of billing accounts retrieved from BillingCenter.

- **Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent's account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle's account as the alternate billing account. The uncle's account is not a subaccount of the parent's account.
You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.
- **Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the `StandAloneBillingSystemPlugin`, the plugin returns a list of sample installments plans. If you are integrated with BillingCenter, the plugin returns a list of valid installment plans for the selected list bill payer. The integration retrieves installments plans from BillingCenter.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the `StandAloneBillingSystemPlugin` or the default integration with BillingCenter, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The `StandAloneBillingSystemPlugin` simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on inte-

grating with a payments system, “Payment Integration” on page 553 in the *Integration Guide*.

If you select **Automatic for Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.
- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

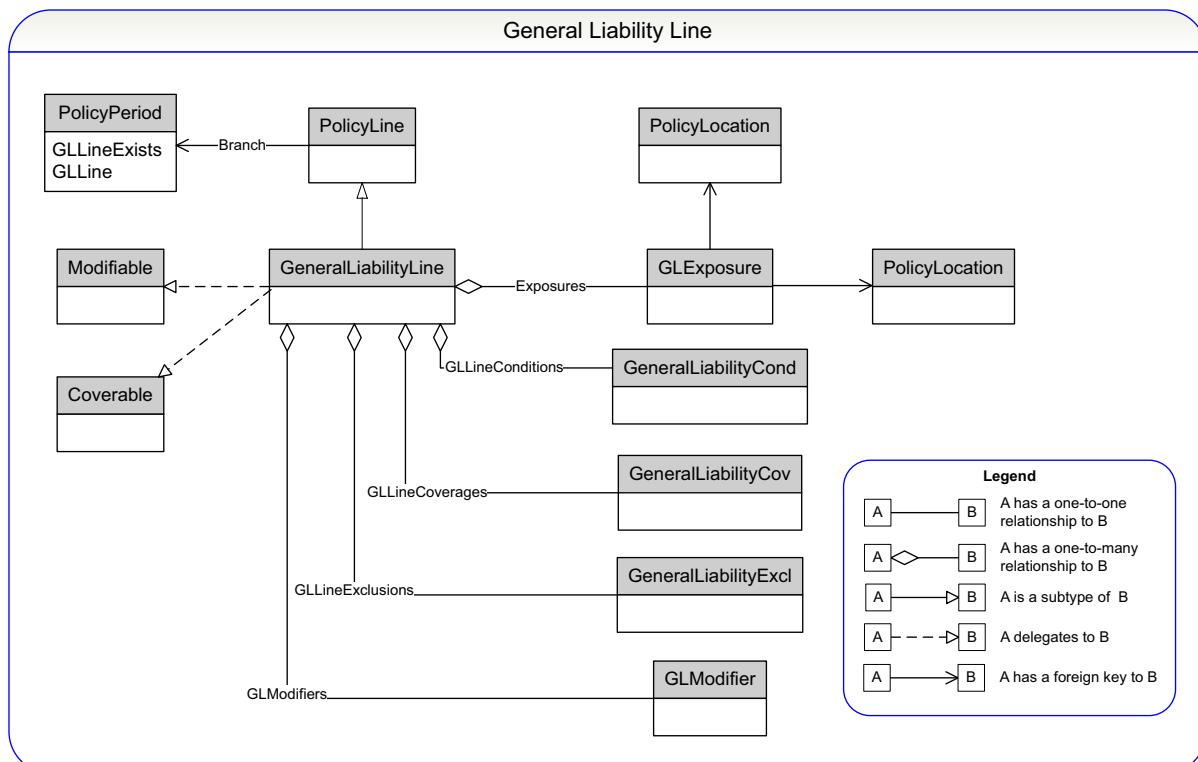
Note: The BillingCenter integration does not support this functionality.

In the **Audit** section, specify whether the policy **Requires final audit**. Your choices are: **Determined By Business Rule**, **Yes**, or **No**.

General Liability Object Model

This section describes the objects or entities associated with the general liability line of business.

PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.



See also

- “Core Entities Associated with Policies” on page 309
- “Cost and Transaction Model for General Liability Line” on page 427

Line Entity in General Liability

The `GeneralLiabilityLine` entity is an entity subtype of `PolicyLine`.

The general liability line delegates to `Coverable`, so you can add coverages to the line. In the default configuration, there are no line-level coverages defined. However, if you add line-level coverages, the `GeneralLiabilityLine` entity has arrays for `GeneralLiabilityCov`, `GeneralLiabilityExcl`, and `GeneralLiabilityCond` that you can use.

The general liability line delegates to `Modifiable`, so you can add modifiers to the line. The `GeneralLiabilityLine` entity has an array of `GLModifier` entities.

Coverage Entity in General Liability

The `GeneralLiabilityCov` entity defines the types of coverage terms that can be assigned for the general liability line. The default configuration provides a number of coverages including condominiums, coverage for injury to leased workers, and designated pollutants. You can view the coverages in Product Designer by navigating to the **Coverages** page in **Product Model** → **Policy Lines** → **General Liability Line**.

Modifier Entity in General Liability

The `GLModifier` entity represents modifiers on the general liability policy line. You can configure modifiers in Product Designer by navigating to **Product Model** → **Products** → **General Liability Line** then going to the **Modifiers** page.

Inland Marine

The inland marine line of business is part of the PolicyCenter base configuration.

Inland marine insurance is a broad type of coverage was developed for shipments that do not involve ocean transport. The insurance covers articles in transit by all forms of land and air transportation as well as bridges, tunnels, and other means of transportation and communication. Floater policies that cover expensive personal items such as fine art and jewelry are included in this category.

Inland marine provides insurance for a wide variety of coverables that:

- Do not have a license plate – therefore, automobiles are not covered by inland marine.
- Do not have a foundation – therefore, buildings are not covered by inland marine.
- Are not considered personal property within a building coverage – therefore, office furniture is not covered by inland marine.

This line of business contains a reference implementation that you can use to accelerate your implementation. This line of business includes reference implementations for jobs, policy file screens, sample rating rules, sample eligibility rules, and forms logic. The reference implementation also provides sample content.

Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can build your own compliance system. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. The default application contains a sample set of these classifications.

This topic includes:

- “Inland Marine Overview” on page 246
- “Inland Marine Screens” on page 246
- “Inland Marine Object Model” on page 254
- “Inland Marine Product Model” on page 256

Inland Marine Overview

The inland marine line provides insurance for separate and distinct types of coverables. These distinct types are referred to as coverage parts. A coverage part is a coverage form that includes an insuring agreement, and therefore, can stand by itself. Coverage parts consist of groups of coverages. Coverages do not extend across coverage parts.

Within PolicyCenter, a carrier defines the coverage parts for which they will provide insurance. A typical carrier might provide insurance for up to 10, 20, or more coverage parts. Within a specific policy, it would be unusual to have insurance for more than four coverage parts.

The default installation of PolicyCenter includes a reference implementation for the following coverage parts:

- **Accounts Receivable** – When records are lost or damaged, accounts receivable provides insurance for the receivable amounts and for the cost to reestablish records. For example, if the records were lost in some calamity, this insurance covers what cannot be collected.
- **Contractors Equipment** – Provides insurance for direct physical loss to portable machinery, equipment and tools used by contractors. This insurance covers both owned and non-owned (care, custody and control) equipment. Equipment rented to others is explicitly excluded. This insurance does not provide coverage for items used in mining, underground or marine projects except when those items are in storage or in open lots.
- **Signs** – Provides insurance for electronic and mechanical signs.

Inland Marine Screens

The inland marine implementation contains a series of screens to choose coverages, assess risk, quote, and select payment options for the policy. This section provides descriptions of fields that define the policy contract and contain related information in the default configuration.

This section describes the following screens in inland marine:

- “Policy Info Screen for Inland Marine” on page 246
- “Coverage Part Selection Screen for Inland Marine” on page 248
- “Buildings and Locations Screen for Inland Marine” on page 248
- “Accounts Receivable Screen for Inland Marine” on page 249
- “Contractors Equipment Screen for Inland Marine” on page 249
- “Signs Screen for Inland Marine” on page 250
- “Risk Analysis Screen for Inland Marine” on page 250
- “Policy Review Screen for Inland Marine” on page 250
- “Quote Screen for Inland Marine” on page 251
- “Forms Screen for Inland Marine” on page 252
- “Payment Screen for Inland Marine” on page 252

Policy Info Screen for Inland Marine

The **Policy Info** screen captures basic information about the policy.

This screen contains information about the primary named insured, producer information, policy details, and selected underwriting company. The buttons and choices that you see on the **Policy Info** screen vary based on the contacts associated with the account and how they are associated with the policy. For example, under **Primary Named Insured** you might see any or all of the following:

- New Company
- New Person

- From Address Book
- Existing Contact

Not all choices appear at any given time.

The following table describes key fields of the **Policy Info** screen.

Name of field	Description
Primary Named Insured	<p>PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select:</p> <ul style="list-style-type: none">• New Company• New Person• From Address Book• Existing Contact <p>If the named insured is a person, the social security number is required in Official IDs. If the named insured is a business, then FEIN is required.</p> <p>For Existing Contact, PolicyCenter lists contacts with the Account Holder or Named Insured role on the account. The list does not include contacts who are already the primary or additional named insureds on this policy.</p>
Organization Type	Enables you to select the type of organization such as whether this business is a corporation or partnership.
Additional Named Insureds	Use to create additional named insureds. This might be a business partner, for example. The Add button enables you to select: <ul style="list-style-type: none">• New Company• New Person• From Address Book• Existing Additional Named Insured• Other Contacts
Policy Details	Includes the Term Type , Effective Date , Expiration Date , Written Date , and the Base State .
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen. Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p> <p>If you change the producer in the middle of a policy period, the new producer is the Producer of Service. The original Producer of Record remains. When the policy is renewed, the Producer of Service becomes the Producer of Record.</p>
Underwriting Companies	<p>The underwriting company is automatically set for a policy version based on logic defined in segmentation classes. You can select a different underwriting company if you have the correct permissions.</p> <p>The drop-down list contains a configurable set of choices.</p>
Preferred Currency	If PolicyCenter is configured as a multicurrency system, the Policy Info screen displays this label and the following fields related to the preferred currency.

Name of field	Description
Coverage	The preferred or default currency for coverages on the policy. The Coverage currency choices come from the policy line configuration in Product Designer. In the base configuration, the default is the preferred coverage currency on the account, Account.PreferredCoverageCurrency.
Settlement	The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the Quote and other screens. In the base configuration you can select one of the following currencies: <ul style="list-style-type: none"> • USD • EUR • GBP • CAD • AUD • RUB • JPY The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, Account.PreferredSettlementCurrency. You can set the preferred settlement currency in the Account File Summary screen Currencies → Settlement field.

Coverage Part Selection Screen for Inland Marine

The **Coverage Part Selection** screen allows you to select coverage parts for the policy. At least one coverage part must be selected to create a valid policy. The default installation provides the following coverage parts:

- Accounts Receivable
- Contractors Equipment
- Signs

When you add a coverage part, a link to the screen for that coverage part appears in the sidebar.

Buildings and Locations Screen for Inland Marine

The **Buildings and Locations** screen allows you to add locations and buildings. By default, PolicyCenter adds the primary account location. (If the primary account location is not associated with the policy, you can remove it from the policy.) You can add new or existing locations and buildings to the policy.

Accounts Receivable Screen for Inland Marine

If you add an Accounts Receivable coverage part in the Coverage Part Selection screen, a link to the Accounts Receivable screen appears in the left sidebar.

Field	Description
Part Level Information	This section displays fields related to the coverage part as a whole.
Reporting	Select whether or not the customer will use monthly reports. (You must configure PolicyCenter to support monthly reporting within this line of business.)
Business Class	Specify the type of business.
Coinurance Pct.	Specify the coinsurance percentage.
Off Premises Property	Specify coverages for accounts receivables that are not located on the premises. You can enter a description of the property and limit.
Account Receivable Coverages	Add covered buildings at a location where you store accounts receivable information. Use the Receptacle Type to specify the type of filing cabinets or safe where the accounts are stored. Select Forwarded to Home Office if account receivable information is duplicated at the home office. Use Percent Duplicated to specify how much of the information is duplicated (whether or not it is forwarded to the home office). The Limit specifies upper limit of how much the insurer will pay.
Excluded Customers	Add customer accounts that will be excluded from coverage.

When multicurrency display is enabled, you can set the currency on a coverable in the user interface. PolicyCenter then copies this coverage currency to each coverage on the coverable. The main screen for each coverable has the following multicurrency field:

Field	Description
Coverages in	Select the currency for the coverages on this coverable. By default, this value is set to the Policy Info → Preferred Currency → Coverage field. The drop-down list contains the Available Coverage Currencies defined in Product Designer on the main screen of the policy line.

See also

- “Multicurrency Policies” on page 517

Contractors Equipment Screen for Inland Marine

If you add Contractors Equipment in the Coverage Part Selection screen, a link to the Contractors Equipment screen appears in the sidebar.

Coverages Tab

The Coverages tab includes the following fields.

Field	Description
Part Level Information	This section displays fields related to the coverage part as a whole.
Contractor Type	Select the type of contractor.
Coinurance	Specify the coinsurance percentage.
Per occurrence limit	Set a limit on the amount of claim per loss. This limit is not tied to a specific coverage.
Reporting	Select whether or not the customer will use monthly reports. (You must configure PolicyCenter to support monthly reporting within this line of business.)
Exclusions	Specify whether theft or vandalism is covered or excluded.

Field	Description
Unscheduled Equipment	Specify limit, deductible, and maximum individual item value for miscellaneous unscheduled items and employee tools. Unscheduled equipment are generally smaller and less valuable items than scheduled equipment.
Part Level Coverages	Add coverage for rented equipment, rental reimbursement, additionally acquired property, debris removal, pollution cleanup, and preservation of property. These coverages are not tied to a specific piece of equipment.
Scheduled Equipment	Add equipment details such as description, manufacturer, model, and model year. Specify equipment coverage limit, deductible, and valuation. Add any additional interests.

Underwriting Information Tab

The **Underwriting Information** tab contains underwriting questions about contractors equipment. In the default configuration, the questions are used to gather information but do not raise any underwriting issues. You can configure the questions to raise underwriting issues. For more information, see “Underwriting Authority” on page 411 and “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*.

Signs Screen for Inland Marine

If you add **Signs** in the **Coverage Part Selection** screen, a link to the **Signs** screen appears in the left sidebar. Select the **Coinsurance** percentage which applies to all signs. For each sign, specify a location, sign type, limit, and deductible.

Risk Analysis Screen for Inland Marine

The **Risk Analysis** screen allows you to enter data that is used to evaluate the risk of the applicant.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – This tab appears when viewing a policy. It does not appear in a work order such as a submission. This tab allows you to add an underwriting referral reason to a policy. Underwriting referral reasons are usually only used when a notable condition arises outside of a job, possibly outside of the data that PolicyCenter maintains on the policy. For more information about this tab, see “Working with Underwriting Referral Reasons” on page 416.
- **UW Issues** – This tab displays underwriting issues that were raised during the job. For more information on this tab, see “Working with Underwriting Issues” on page 414.
- **Prior Policies** – This tab displays information about prior policies usually with another carrier. You can add information about prior policies.
- **Claims** – This tab allows you to search for loss claims on a related policy or by loss date. When integrated with a claim system, this tab displays claims from the claim system. For more information, see “Viewing Loss Claims for Policies” on page 716.
- **Prior Losses** – This tab displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

In the default installation, underwriting issues are configured only for rating overrides. Rating overrides allow you to manually override the premium that the rating engine automatically generates for a policy. For more information, see “Rating Overrides” on page 445.

Policy Review Screen for Inland Marine

The **Policy Review** screen is similar to the summary screen for other lines of business.

For a submission job, this screen contains all the policy data in summary form. For other jobs, this screen displays the differences between the policy versions. It is useful to review this screen before generating a quote.

If any of the information needs to be changed, click the menu link on the left sidebar to edit the appropriate screen.

This screen contains tabs for each coverage part added to the policy. Each tab contains a summary of the selected coverages.

If PolicyCenter is configured as a multicurrency system, the **Policy Review** screen displays the cost for each coverable's coverages in the currency set on the coverable.

Quote Screen for Inland Marine

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen.

This screen is similar to the quote screen for other lines of business. It contains the basic information at the top including the total premium and taxes and surcharges. The bottom of the screen displays a breakdown of the premium.

Within the **Quote** screen, your choices at this point include:

- Editing the policy (and thereby invalidating the current quote)
- Creating a new version of the policy
- Saving a draft of the policy
- Binding or issuing the policy
- Withdrawing, declining, or not taking the policy
- Printing the policy

PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

In addition, Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

If you have Guidewire Rating Management, the **Policy Premium** tab has a **Show Rate Worksheet** button. For more information about this button, see “Rating Worksheets” on page 548 in the *Application Guide*.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Policy Premium Tab

The **Override Rating** button allows you to manually override the premium that the rating engine automatically generates for a policy. For more information, see “Rating Overrides” on page 445.

The **Compact View** and **Extended View** buttons display the rating information in compact or extended form. If the approximate page length is greater than 50, then PolicyCenter displays the compact view. Otherwise, PolicyCenter displays the extended view. The compact view shows summary information on the initial screen and allows the user to click to view details. The extended view displays the details on the initial screen.

Gosu code in the PCF file determines which view to display initially. The panel set that contains the **Compact View** and **Extended View** buttons is `RatingCumulDetailsPanelSet.IMLine.pcf`.

This PCF file includes a modal panel set. The `initialValue` of the `pageLength` variable determines whether the modal panel set displays initially in drill-down or scroll mode. At top of this screen, click the panel set for this file to display the **Variables** and **Code** tabs at the bottom of the screen. On the **Variables** tab, the `pageLength` variable calculates an `initialValue`. The **Code** tab sets the view mode based on the value of the `pageLength` variable.

Forms Screen for Inland Marine

The **Forms** screen allows you to associate forms with a policy.

PolicyCenter does not store the content of any forms associated with a policy. Therefore, the **Forms** screen only identifies that zero, one, or more, forms have been associated with the policy. The **Forms** screen simply lists form instances that indicate the physical forms, which are usually printed by an issuance system. Each form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form. The form content is not stored within PolicyCenter.

When you click **Quote** or **Bind**, PolicyCenter infers which forms the policy needs. In the default application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy Forms” on page 461
- “Policy Form Pattern Administration” on page 683 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment Screen for Inland Marine

The **Payment** screen displays payment information for the policy.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- **Total Premium**
- **Taxes and fees**
- **Total Cost** – The sum of the previous fields.

Billing

Under the **Billing** heading, you can select:

- **Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- **Alt Billing Account** – Select an alternate billing account.
- **Billing Contact** – Select a person or company as the billing contact.

Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- **List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer's mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample billing accounts. If you are integrated with **BillingCenter**, the plugin returns a list of billing accounts retrieved from **BillingCenter**.

- **Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent's account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle's account as the alternate billing account. The uncle's account is not a subaccount of the parent's account.

You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.

- **Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample installments plans. If you are integrated with **BillingCenter**, the plugin returns a list of valid installments plans for the selected list bill payer. The integration retrieves installments plans from **BillingCenter**.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the `StandAloneBillingSystemPlugin` or the default integration with BillingCenter, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The `StandAloneBillingSystemPlugin` simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payments system, “Payment Integration” on page 553 in the *Integration Guide*.

If you select **Automatic** for **Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.
- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

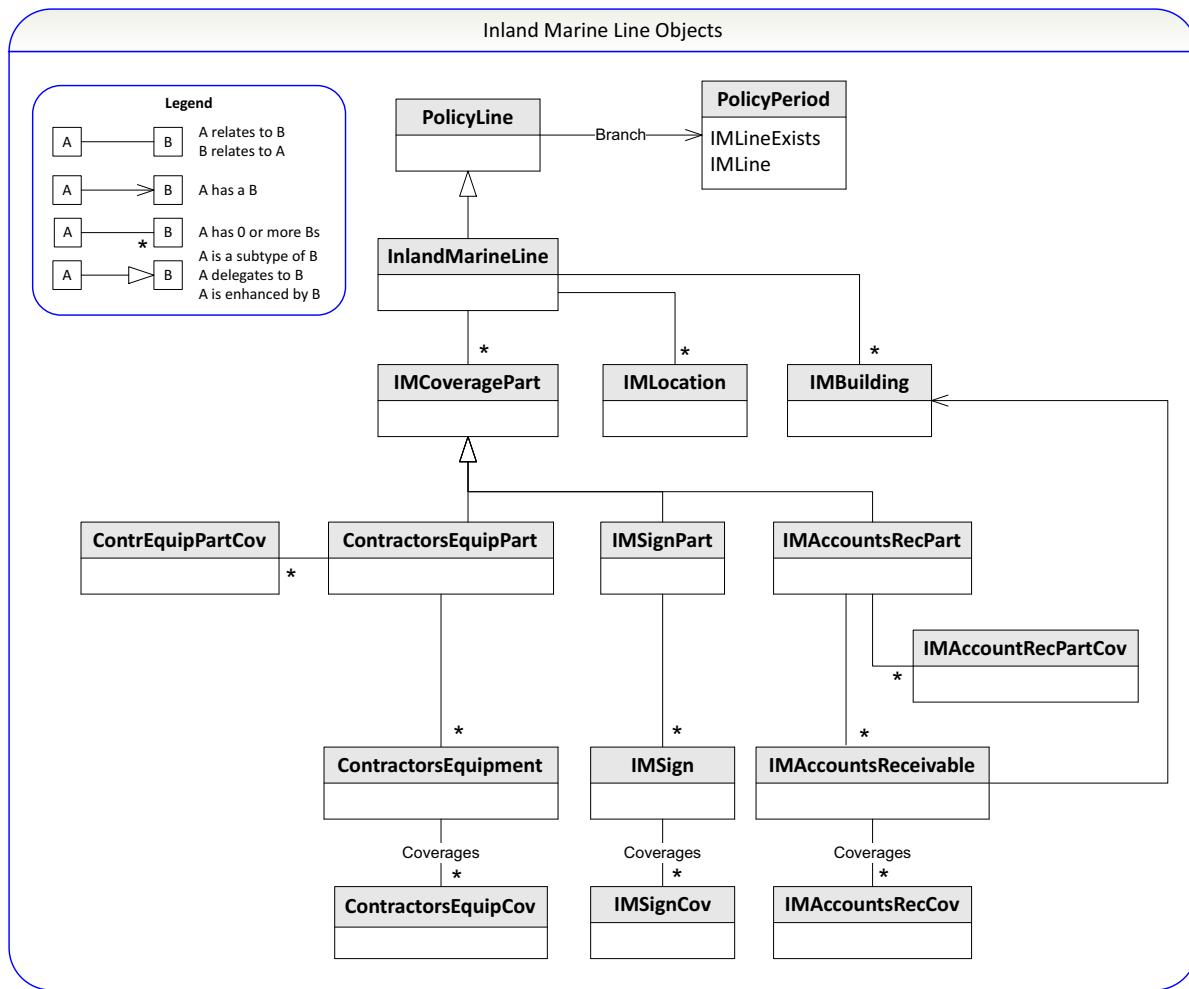
Note: The BillingCenter integration does not support this functionality.

Inland Marine Object Model

This section describes the objects or entities associated with the inland marine line of business.

PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.

The following diagram of the inland marine object model shows how some of the key entities relate to each other.



The inland marine line is composed of three coverage parts: one each for accounts receivable, contractors equipment, and signs. Coverage parts are composed of coverages and coverable objects. This model of coverage parts is unique to the inland marine line. Most of the lines of business are directly composed of coverages and coverable objects.

The **InlandMarineLine** entity has an array key to **IMCoveragePart**. The abstract **IMCoveragePart** is not a coverable, but its **ContractorsEquipPart**, **IMAccountsRecPart**, and **IMSignPart** subtypes are.

The sign coverage part, **IMSignPart**, has no part level coverages. Coverages for signs are on the **IMSign** entity. This coverage part uses information about locations. It does not use information about buildings.

The contractor's equipment coverage part, **ContractorsEquipPart**, has coverages on both the coverage part and the equipment. This coverage part does not use information about buildings or locations.

The accounts receivable coverage part, **IMAccountsRecPart**, is the most common type. This coverage part has part level coverages (**IMAccountRecPartCov**) and coverages by receptacle type (**IMAccountsReceivable**) within each building. This coverage part uses information about buildings and locations.

See also

- “Core Entities Associated with Policies” on page 309
- “Cost and Transaction Model for Inland Marine Line” on page 430

Inland Marine Product Model

Various features of inland marine are defined in the product model.

The product model is the PolicyCenter feature that identifies the different types of policies, or products, that a given instance of PolicyCenter offers. For each product, the product model details all of the choices around what can be covered. You can think of the product model as a product configuration.

In Studio, the following files in the **Product Model** folder are related to inland marine:

- **Inland Marine** in the **Products** folder
- **Inland Marine Line** in the **Policy Lines** folder
- **IM Contractors Equipment Question** in the **Question Sets** folder

See also

- “Configuring the Product Model” on page 13 in the *Product Designer Guide*

Personal Auto

The personal auto line of business is part of the PolicyCenter base configuration.

This line of business contains a reference implementation that you can use to accelerate your implementation. This line of business includes reference implementations for jobs, policy file screens, sample rating rules, sample eligibility rules, and forms logic. The reference implementation also provides sample content.

Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can build your own compliance system. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. The default application contains a sample set of these classifications.

This topic includes:

- “Personal Auto Overview” on page 257
- “Working with Personal Auto” on page 260
- “Personal Auto Screens” on page 260
- “Personal Auto Object Model” on page 269
- “Configuring Personal Auto” on page 275

Personal Auto Overview

Most people are somewhat familiar with personal auto based on their experiences of owning a vehicle and needing insurance. Generally they know that certain parameters affect the cost of the policy, such as:

- The type of vehicle insured
- Where the driver lives
- How far the vehicle is driven in a year
- The driving record
- The gender and age of the driver
- The available discounts

- The deductible amount
- The liability limits

There are other factors that can be taken into account, such as whether the person applying for the policy already has a policy with the same carrier. During the submission process, PolicyCenter captures this information and passes it to a rating engine, which, in turn, uses the information to generate a quote. Within PolicyCenter, the **Policy Info** screen (shown in the following submission example) begins the initial capture of this information that is critical in obtaining a quote.

Policy Info	
Primary Named Insured	
Name	Ray Newton
Phone	818-446-1206
Address	1253 Paloma Ave Floor 0000 Developer Unit Habitation Cube #0000 Arcadia, CA 91007
County	United States San Mateo
Address Type	Home
Address Description	Created by the Address Builder with code 0
Official IDs	
SSN	342-56-8729
Secondary Named Insured	

Policy Details	
Term Type	Other
Term Number	2
Effective Date	08/13/2013
Expiration Date	01/13/2014
Rate as of Date	08/13/2013 1:20:24 PM PDT
Producer of Record	
Organization	Armstrong and Company
Producer Code	100-002541 Armstrong (Premier)
Producer of Service	
Organization	Armstrong and Company
Producer Code	100-002541 Armstrong (Premier)

1. Contains key information at a glance. Click **Account #** to jump to the **Account File Summary** screen where you can make edits.
2. The highlighted item in the left sidebar indicates the screen that you are viewing.

See also

- “Side-by-side Quoting” on page 159

Motor Vehicle Records in Personal Auto

A motor vehicle record (MVR) documents a driver’s driving history. The MVR report contains information such as identifying data, license status, convictions, traffic violations, accidents, license suspensions, and revocations. In the U.S., the information in this report usually comes from the Department of Motor Vehicles (DMV) for each jurisdiction. The information in the report can vary by jurisdiction. In the U.S., most service providers provide MVR data for all jurisdictions, so that you only need to integrate with a single service provider.

A carrier uses the MVR to evaluate the risks associated with a given driver. Violations are assigned point values, with more severe violations having a higher point value. A high MVR point total indicates a high risk driver and can result in higher policy premiums.

In PolicyCenter, the personal auto line of business provides an MVR integration for the U.S market. The default configuration provides an integration to a demonstration version of a service provider that simulates receiving MVR reports for selected drivers.

You can configure PolicyCenter to integrate with the service provider of your choice. You can extend the MVR integration to other lines of business, such as commercial auto. You can extend the MVR integration to other countries than the U.S.

See also

- “Motor Vehicle Record (MVR) Plugin” on page 245 in the *Integration Guide*

MVR Implementation Overview

The MVR implementation in PolicyCenter provides:

- Timely availability of MVR data for performing risk analysis and for rating
- Elimination of the often manual lookup of MVR data in a secondary system

Since there is a cost associated with retrieving data from the external MVR service provider, PolicyCenter stores the retrieved MVR data to minimize future lookups. PolicyCenter requests the MVR report from an external service provider only if it does not have a copy or if the report is considered to be *stale*.

The retrieved MVR report is maintained in a separate MVR repository that is independent of the driver's account or policy. This allows the MVR report to be reused for drivers associated with multiple accounts or policies. This also provides flexibility as to when to update an MVR report on in-force policies. In the default configuration, the MVR report for an in-force policy is updated at renewal or during a policy change.

Accounts maintain a subset of the MVR data. In the default configuration, this subset consists of accidents and violations. Accounts use this subset to:

- Store the most current values for accidents and violations
- Provide current values for submissions and renewals

Policies also maintain a subset of MVR data. In the default configuration, this subset consists of accidents and violations. Policies use this subset to:

- Rate the policy
- Trigger an underwriting issue
- Allow the user to override the MVR values for accidents and violations

Accidents and Violations

In PolicyCenter, you can set the number of accidents and violations at the account and policy levels. You set these values in a policy transaction on the **Driver Details → Roles** tab. You might set these values if the MVR report:

- Is not yet available.
- Has inaccurate information.
- Does not contain an accident or violation that occurred recently.

For example, in a quick quote or when the MVR report is requested at bind time, the agent can ask the applicant for the number of accidents and violations. This information is needed for generating a quote. Later, when the MVR report is received, the applicant's information is validated against the report. If there is a discrepancy, PolicyCenter raises an underwriting issue that prevents binding the policy period. The agent can also choose **Do Not Order MVR** on the **Driver Details → Roles** tab if she knows that the MVR report does not exist or is inaccurate.

PolicyCenter propagates account values for accidents and violations to new policy transactions that create a new policy term, specifically submissions, renewals, and rewrites. PolicyCenter copies the number of accidents and violations on the account to a new policy term when a draft is created. For example, a new submission copies the number of accidents and violations stored on the account, while a policy change does not. In a policy change, the number of accidents and violations on the policy is not updated from the account. This is because the rating information cannot change on a policy term that has been bound.

See also

- “Motor Vehicle Record Object Model” on page 272
- “Configuring Personal Auto Motor Vehicle Records” on page 275
- “Account File Contacts Screen” on page 326 for MVR data stored with account contacts that have the driver role.

Working with Personal Auto

This topic contains step-by-step instructions for working with a personal auto policy in the user interface.

Copying Coverages to Other Vehicles

You can copy coverages from one vehicle to other vehicles in the policy. This copies the full coverage pattern on the vehicle. For example, if a coverage is selected on the copy from vehicle, that coverage is copied to all copy to vehicles. In addition, if a coverage is not selected on the copy from vehicle, that lack of selection is copied to all the copy to vehicles.

Product model synchronization removes coverages that are not available in the destination risk.

1. In a personal auto policy, navigate to the **PA Coverages** screen.
2. Under the heading **Coverages applied per vehicle**, select **Copy Coverages**. This button is available if there are at least two vehicles on the policy.
3. On the **Copy Coverages** screen, select a vehicle from the **Copy From Vehicle** drop-down list.
4. Under the heading **Copy To**, select one or more vehicles to copy coverages to, or click **Copy To All** at the top of the screen.

See also

- “Configuring Copy Coverages for Personal Auto” on page 275

Personal Auto Screens

This section describes the types of information captured in personal auto submission wizard steps that may have implications in the policy. The main sections of the wizard include the **Policy Info**, **Drivers**, **Vehicles**, and **PA Coverages** screens.

To select the personal auto policy product in the **New Submission** screen, you must select the following:

- A personal account (as opposed to a company account).
- The type of submission (**Quick Quote** or **Full Application**)

This section describes the following screens in personal auto:

- “Policy Info Screen for Personal Auto” on page 260
- “Drivers Screen for Personal Auto” on page 262
- “Vehicles Screen for Personal Auto” on page 263
- “PA Coverages Screen for Personal Auto” on page 264
- “Risk Analysis Screen for Personal Auto” on page 265
- “Policy Review Screen for Personal Auto” on page 265
- “Quote Screen for Personal Auto” on page 265
- “Forms Screen for Personal Auto” on page 266
- “Payment Screen for Personal Auto” on page 266

Policy Info Screen for Personal Auto

The **Policy Info** screen captures basic information about the policy.

This screen contains information about the primary named insured, producer information, policy details, and selected underwriting company. The buttons and choices that you see on the **Policy Info** screen vary based on the

contacts associated with the account and how they are associated with the policy. For example, under **Primary Named Insured** you might see any or all of the following:

- New Company
- New Person
- From Address Book
- Existing Contact

Not all choices appear at any given time.

The following table describes key fields of the **Policy Info** screen.

Name of field	Description
Primary Named Insured	<p>PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select:</p> <ul style="list-style-type: none">• New Person• From Address Book• Existing Contact <p>For Existing Contact, PolicyCenter lists contacts with the Account Holder or Named Insured role on the account. The list does not include contacts who are the primary or additional named insureds on this policy.</p>
Secondary Named Insured	<p>Use to create the secondary named insured. This might be a spouse or a child, for example. The Add button enables you to select:</p> <ul style="list-style-type: none">• Add Existing• New Person• New Contact from Address Book
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p> <p>If you change the producer in the middle of a policy period, the new producer is the Producer of Service. The original Producer of Record remains. When the policy is renewed, the Producer of Service becomes the Producer of Record.</p>
Underwriting Companies	<p>The underwriting company is automatically set for a policy version based on logic defined in segmentation classes. You can select a different underwriting company if you have the correct permissions.</p> <p>The drop-down list contains a configurable set of choices.</p>
Preferred Currency	If PolicyCenter is configured as a multicurrency system, the Policy Info screen displays this label and the following fields related to the preferred currency.

Name of field	Description
Coverage	The preferred or default currency for coverages on the policy. The Coverage currency choices come from the policy line configuration in Product Designer. In the base configuration, the default is the preferred coverage currency on the account, Account.PreferredCoverageCurrency.
Settlement	The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the Quote and other screens. In the base configuration you can select one of the following currencies: <ul style="list-style-type: none"> • USD • EUR • GBP • CAD • AUD • RUB • JPY The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, Account.PreferredSettlementCurrency. You can set the preferred settlement currency in the Account File Summary screen Currencies → Settlement field.

Drivers Screen for Personal Auto

Use this screen to define the driver or drivers of any vehicles that this policy covers. It is initially blank. Select **Add** to define any vehicle drivers that you want to associate with this policy. PolicyCenter automatically furnishes the names of any contacts that have one of the following associations on the account:

- Named insureds
- Drivers
- Account holders

For example, if Joe Smith is a driver on another policy on this account but has not been added to this policy, he appears as a choice under **Add Existing**.

This screen also contains information on the driver that the rating system can use to modify the premium. Examples of this information are: date completed training class, number of accidents and violations, year first licensed.

Note: You must add one or more drivers before you add vehicles to the policy.

Roles Tab

The Roles tab displays information such as:

- Date Completed Training Class
- Year First Licensed
- Qualifies for a Good Driver Discount
- Do Not Order MVR

In the **Accident/Violation Summary** at the bottom of the screen, you can enter **Number of Accidents** and **Number of Violations** at either the **Policy Level** or **Account Level**. The final column of this summary displays the values for accidents and violations from the **MVR Report**.

In a new submission, renewal, and rewrite, the number of accidents and violations is set to the account values for each driver. Therefore, the most up-to-date information is applied to the policy at the start of these jobs, and can be updated by the underwriter if required.

Motor Vehicle Record Tab

The **Motor Vehicle Record** tab displays summary information for the selected driver. The summary information includes violations and accidents and the point value for each. Points can affect the policy premium. However, they do not affect the premium in the default configuration. Points can also provide information to the underwriter. A certain number of points may lead to suspended license.

Click **MVR Report Details** to view the complete report. PolicyCenter displays the **Motor Vehicle Records** screen. For each record, this screen displays the complete motor vehicle record on the following tabs:

- **Driver** – Information about the driver such as name, date of birth, gender, address.
- **Incidents** – Select an incident to display its details.
- **Additional Info** – Miscellaneous information returned by the service provider.

Vehicles Screen for Personal Auto

In the **Vehicles** screen, you create the vehicle or vehicles that this policy is to insure. You can specify basic vehicle information such as where the vehicle is garaged and who is assigned to drive the vehicle. In the optional **Additional Interest** tab, you can provide the name of the institution that owns the vehicle until it is paid off by the insured.

There are two buttons, **Create Vehicle** and **Remove Vehicle**. Clicking **Create Vehicle** displays two cards: **Vehicle Details** and **Additional Interest**. Use the **Vehicle Details** card to enter basic information about a vehicle. For example:

Field name	Choices	Description
VIN	Enter the number.	This is an integration point. In the development environment, PolicyCenter supplies a demonstration plugin. In a production environment, this field would likely link to a working plugin that retrieves vehicle data based on the VIN number. See "PolicyCenter Integration Points" on page 34 for additional information.
Location Name	The default in the Where Garaged section drop-down list is the primary account location.	Select the down arrow button to the right of the field to: <ul style="list-style-type: none">• Edit the current garage location• Change to a different account location• Create another garage location After entering the address, PolicyCenter automatically fills in the personal auto territory code. If PolicyCenter cannot uniquely determine the code, then you can manually enter it or find another by searching. You may edit the territory code by choosing Edit Location to edit information about the garaging location. Changing the location updates the account location, but not other policies that are in-force. All vehicles must be garaged in the same jurisdiction, otherwise PolicyCenter generates an error.
Assign Drivers to Vehicles section	You must add (or optionally remove) a driver.	The driving total for all drivers must equal 100 percent. You can only add drivers that have already been added on the Drivers screen.
Vehicle Rate Modifiers	Only select if a modifier applies to this vehicle.	Modifiers affect the policy premium calculated during the quote process.

When multicurrency display is enabled, you can set the currency on a coverable in the user interface. PolicyCenter then copies this coverage currency to each coverage on the coverable. The main screen for each coverable has the following multicurrency field:

Field	Description
Coverages in	Select the currency for the coverages on this coverable. By default, this value is set to the Policy Info → Preferred Currency → Coverage field. The drop-down list contains the Available Coverage Currencies defined in Product Designer on the main screen of the policy line.

See also

- “Multicurrency Policies” on page 517

PA Coverages Screen for Personal Auto

Note: The PolicyCenter user interface for personal auto has been designed to support only a single jurisdiction in a policy. This reflects the convention used by most carriers regarding auto policies.

The personal auto coverage screen displays both required and optional coverages. Some coverages apply to all vehicles in the jurisdiction while other coverages apply for each vehicle. For example, auto liability and medical payments apply to all vehicles in the jurisdiction. To remove a coverage from the policy, clear the checkbox for that coverage.

Some coverages are required and cannot be deselected. Others are selected but you can deselect them. PolicyCenter organizes coverages into the following selection types:

Type	Description
Required	Required (by a jurisdiction, for example). These check boxes are selected and you cannot clear them.
Suggested	PolicyCenter suggests that these coverages are appropriate for the policy. They are initially selected, but you can deselect them.
Electable	PolicyCenter generally does not display these coverages initially. You must search for this type of coverage on the Additional Coverages card. Only at this point does PolicyCenter display the coverage, which you can now select. However, certain electable coverages are in coverage categories that PolicyCenter always includes on the PCF. PolicyCenter displays these coverages initially, but leaves them unselected. “Towing and Labor” is an example of this kind of electable coverage.

Coverages Applied per Vehicle

In certain jurisdictions, if you have multiple vehicles, you can select basic coverages that apply to all vehicles, and then select additional coverages that apply to each vehicle. For example, you may want to have towing and labor coverage on your commuter car but not on the secondary car that your teenage child drives. The **Coverages applied per vehicle** section displays each vehicle and its associated coverages.

You can use the **Copy Coverages** button to copy coverages from one vehicle to others. For step-by-step instructions, see “Copying Coverages to Other Vehicles” on page 260.

Additional Coverages Tab

You can use the **Additional Coverages** tab to choose other coverages that are less commonly selected, such as electronic equipment.

Risk Analysis Screen for Personal Auto

The Risk Analysis screen allows you to enter data that is used to evaluate the risk of the applicant.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – This tab appears when viewing a policy. It does not appear in a work order such as a submission. This tab allows you to add an underwriting referral reason to a policy. Underwriting referral reasons are usually only used when a notable condition arises outside of a job, possibly outside of the data that PolicyCenter maintains on the policy. For more information about this tab, see “Working with Underwriting Referral Reasons” on page 416.
- **UW Issues** – This tab displays underwriting issues that were raised during the job. For more information on this tab, see “Working with Underwriting Issues” on page 414.
- **Prior Policies** – This tab displays information about prior policies usually with another carrier. You can add information about prior policies.
- **Claims** – This tab allows you to search for loss claims on a related policy or by loss date. When integrated with a claim system, this tab displays claims from the claim system. For more information, see “Viewing Loss Claims for Policies” on page 716.
- **Prior Losses** – This tab displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.
- **Motor Vehicle Records** – For each driver on the policy, this tab displays:
 - MVR status
 - Summary data such as accidents, violations, and points
 - Do Not Order MVR status

Click **MVR Report Details** to view to complete report. This tab displays the MVR reports after completing the **All ordered MVRs received** activity.

Policy Review Screen for Personal Auto

The Policy Review screen is similar to the summary screen for other lines of business.

For a submission job, this screen contains all the policy data in summary form. For other jobs, this screen displays the differences between the policy versions. It is useful to review this screen before generating a quote. If any of the information needs to be changed, click the menu link on the left sidebar to edit the appropriate screen.

For a submission job, this screen includes jurisdictional level coverages that apply to all vehicles and those that apply to separate vehicles.

If PolicyCenter is configured as a multicurrency system, the Policy Review screen displays the cost for each coverable's coverages in the currency set on the coverable.

Quote Screen for Personal Auto

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen.

This screen is similar to the quote screen for other lines of business. It contains the basic information at the top including the total premium and taxes and surcharges. The bottom of the screen displays a breakdown of the premium.

Within the **Quote** screen, your choices at this point include:

- Editing the policy (and thereby invalidating the current quote)
- Creating a new version of the policy

- Saving a draft of the policy
- Binding or issuing the policy
- Withdrawing, declining, or not taking the policy
- Printing the policy

PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

In addition, Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

If you have Guidewire Rating Management, the **Policy Premium** tab has a **Show Rate Worksheet** button. For more information about this button, see “Rating Worksheets” on page 548 in the *Application Guide*.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Forms Screen for Personal Auto

The **Forms** screen allows you to associate forms with a policy.

PolicyCenter does not store the content of any forms associated with a policy. Therefore, the **Forms** screen only identifies that zero, one, or more, forms have been associated with the policy. The **Forms** screen simply lists form instances that indicate the physical forms, which are usually printed by an issuance system. Each form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form. The form content is not stored within PolicyCenter.

When you click **Quote** or **Bind**, PolicyCenter infers which forms the policy needs. In the default application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy Forms” on page 461
- “Policy Form Pattern Administration” on page 683 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment Screen for Personal Auto

The **Payment** screen displays payment information for the policy.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- Total Premium**
- Taxes and fees**
- Total Cost** – The sum of the previous fields.

Billing

Under the **Billing** heading, you can select:

- Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- Alt Billing Account** – Select an alternate billing account.
- Billing Contact** – Select a person or company as the billing contact.

Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer’s mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample billing accounts. If you are integrated with BillingCenter, the plugin returns a list of billing accounts retrieved from BillingCenter.

- Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent’s account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle’s account as the alternate billing account. The uncle’s account is not a subaccount of the parent’s account.

You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.

- Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the `StandAloneBillingSystemPlugin`, the plugin returns a list of sample installments plans. If you are integrated with BillingCenter, the plugin returns a list of valid installment plans for the selected list bill payer. The integration retrieves installments plans from BillingCenter.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the `StandAloneBillingSystemPlugin` or the default integration with BillingCenter, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The `StandAloneBillingSystemPlugin` simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payments system, “Payment Integration” on page 553 in the *Integration Guide*.

If you select **Automatic** for **Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.
- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

Note: The BillingCenter integration does not support this functionality.

Personal Auto Object Model

This section describes the objects or entities associated with the personal auto line of business.

PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.

This topic includes:

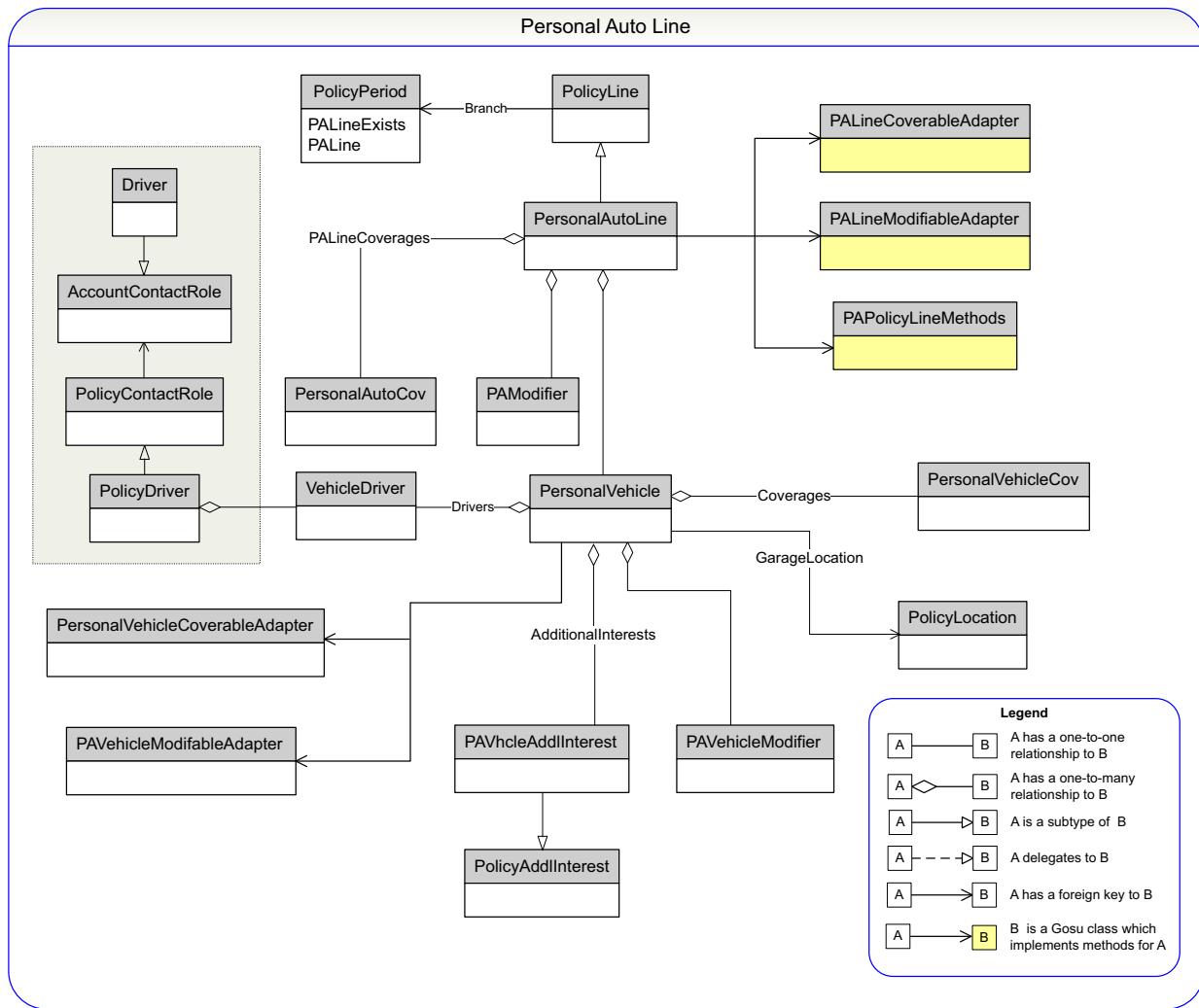
- “Personal Auto Object Model Overview” on page 270
- “PersonalVehicle Entity” on page 270
- “VehicleDriver Entity” on page 271
- “PolicyDriver Entity” on page 271
- “Driver Entity” on page 271
- “Coverages” on page 271
- “Modifiers” on page 272
- “Motor Vehicle Record Object Model” on page 272

See also

- “Core Entities Associated with Policies” on page 309
- “Cost and Transaction Model for Personal Auto Line” on page 431

Personal Auto Object Model Overview

The following diagram of the personal auto object model shows how some of the main personal auto entities relate to each other.



The **PolicyLine** entity contains several subtypes, one of which is **PersonalAutoLine**. (Each line of business is a subtype of **PolicyLine**.) The object model diagram displays relationships which may not be readily apparent. For example, the objects in the shaded portion of the diagram represent a role on the policy, each entity holds portions of the information needed in constructing the policy.

PersonalVehicle Entity

The main entity for personal auto is **PersonalVehicle**, which is a coverable. The **PersonalVehicle** entity tracks items such as VIN numbers, the make, model, color, and type of vehicle. The personal vehicle can be a green 2007 Toyota Camry with a particular VIN number. Notice that **PersonalVehicle** links to the **VehicleDriver** entity in a one-to-many relationship. This indicates that multiple drivers can be associated with (drive) the same vehicle.

A number of arrays that store additional information link to this entity. The following table lists some of these arrays. For complete information, see the PolicyCenter *Data Dictionary*.

Array	Stores information on
AdditionalInterest	Third parties with an additional interest in the vehicle (for example, a bank)
Coverages	All coverages that apply directly to this vehicle
Drivers	All drivers associated with this vehicle
PAVehicleModifiers	Rating information that can affect the premium quote for this vehicle

Note: A coverable must implement the `Coverable` interface. In brief, a *coverable* is an exposure to risk that can be protected by the policy. A coverable may be a tangible property item, a location, jurisdiction, or the policy itself. Coverages attach only to coverables. For information on what constitutes a coverable, see “Covables, Exclusions, Conditions, and Covables Overview” on page 473.

VehicleDriver Entity

The `VehicleDriver` entity is functionally a join table between `PolicyDriver` and `PersonalVehicle`. It contains one record for each driver on a vehicle. The `VehicleDriver` entity links to both the `PersonalVehicle` entity and the `PolicyDriver` entity in a one-to-many relationship. If multiple drivers drive one car, then multiple drivers are associated with that vehicle.

The `VehicleDriver` entity includes, for example, such information as the following:

Field	Description
PrimaryDriver	Indicates whether this driver is primary for the given vehicle or not (used by rating)
PercentageDriven	The percentage this driver drives the vehicle

PolicyDriver Entity

The `PolicyDriver` entity (subtype of `PolicyContact`) contains the array of vehicle drivers. The relationship between the `PolicyDriver` entity, the `VehicleDriver` entity, and the `PersonalVehicle` entity can then be stated, for example as the following:

John Smith is the primary driver of a green 2007 Toyota Camry with VIN number 12345.

The `PolicyDriver` entity also contains the current `ApplicableGoodDriverDiscount`. This is the driver discount that applies to this policy. This is not the same as whether the driver currently qualifies for a good driver discount. The `GoodDriverDiscount` is on the `Driver` entity.

Driver Entity

The `Driver` entity contains information such as driver training, number of accidents and violations, and indicates whether this is the primary driver. It also indicates whether the driver currently qualifies for a good driver discount. Most of the information related to the driver comes from the `Driver` (account contact) entity. The `Driver` entity, in conjunction with the other entities in the shaded portion of the personal auto line diagram, stores the driver information on the policy.

Coverages

A coverage can be defined as a protection from a specific risk. A coverage entity must implement the `Coverage` interface. Coverages always attach to a coverable. There are two types of coverages: property and liability. For example, on an auto policy, a collision property coverage protects the insured’s vehicle and a liability coverage protects the driver for damage done to someone else’s vehicle.

In the base configuration, the personal auto policy line has two types of coverages:

Coverage type	Attaches to	Description
PersonalAutoCov	PersonalAutoLine	Coverage choices that apply to all vehicles in that policy, such as liability coverage.
PersonalVehicleCov	PersonalVehicle	Coverage choices that apply to a specific vehicle, such as a comprehensive deductible or collision information.

Modifiers

A modifier is a value used by the rating engine to adjust the policy premium or some portion of the premium. Modifiers capture information relevant to the pricing of a policy that are not necessarily tied to a specific coverable or coverage. In personal auto, there are the following types of modifiers:

Modifier type	Applies to	Description
PAModifier	The entire policy	A modifier of the policy line. Multi-policy discount or no-loss discount.
PAVehicleModifier	A specific vehicle	A modifier of the vehicle. Premium discounts for such things as ABS (anti-locking brakes), passive restraint, or anti-theft devices.

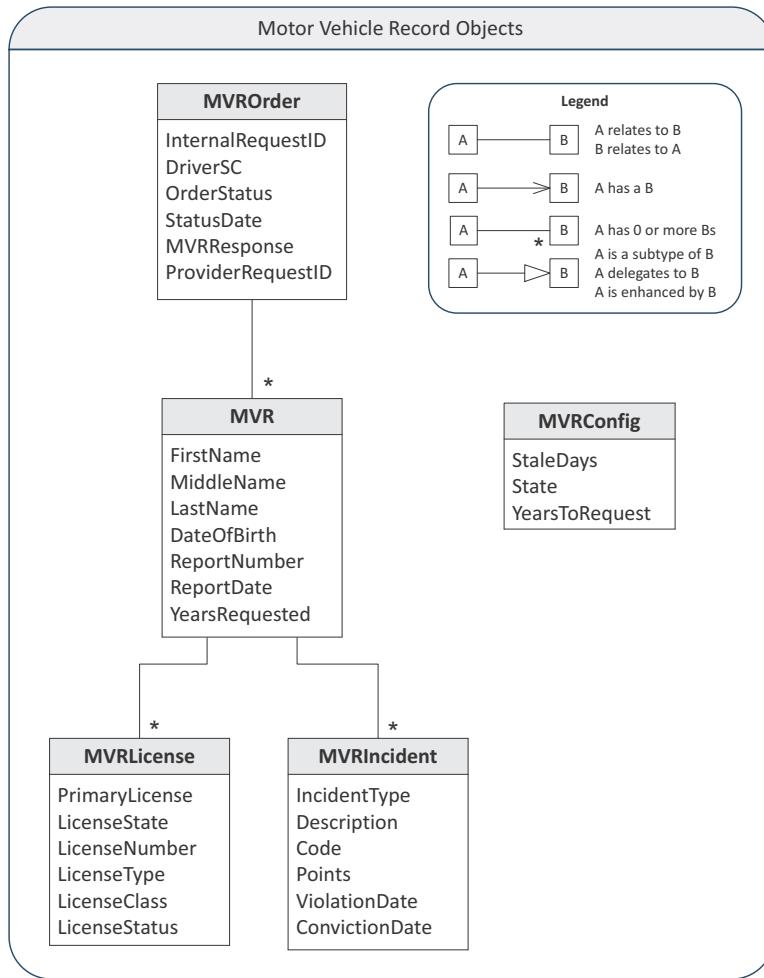
To learn more about how costs work in the personal auto line, see “Quoting and Rating” on page 419.

Motor Vehicle Record Object Model

Objects associated with motor vehicle records occur in three places in PolicyCenter:

- At the system level
- At the account level
- At the policy level

The following illustration shows the object model for entities related to motor vehicle records at the system level. These are the entities created when the agent orders and receives an MVR report from the service provider.



Motor Vehicle Record Order

A **MVROrder** is created when the agent requests an MVR report. PolicyCenter uses this order to manage the status of the request. The **MVROrder** contains the driver's data requested by the service provider for processing the MVR report. PolicyCenter generates an internal request id to uniquely identify each request in the system. The **ProviderRequestID** field stores the request id assigned by service provider.

Motor Vehicle Record

The **MVR** entity stores the MVR report received from the service provider. The **MVR** entity has fields for social security number, gender, and other information identifying the driver.

Different policies, and even accounts, can use the same MVR report. This is the only full copy of the MVR report. Policies and accounts that use this MVR report contain a subset of this data.

Motor Vehicle Record Incidents

The **MVR** can have one or more incidents. The **MVRIncident** entity represents an incident on the **MVR**. The incidents are grouped by **MVRIncidentType** and provide input to the rating engine. Incidents types include accidents, violations, convictions, warnings, among others.

Motor Vehicle Record Licenses

The MVR entity has an array to one or more MVRLicense entities. The MVR report may contain licenses of various types, such as the primary license, a prior license, commercial license, or other type of license.

Linking Motor Vehicle Record to Account and Policy Records

At the system level, each MVROrder has an InternalRequestID field.

At the policy level, the PolicyDriverMVR object has a matching InternalRequestID which links it to the system level MVROrder entity.

Account level MVR data matches the MVR system data based on account search criteria. Account level MVR data matches MVR system data if the following fields have the same values:

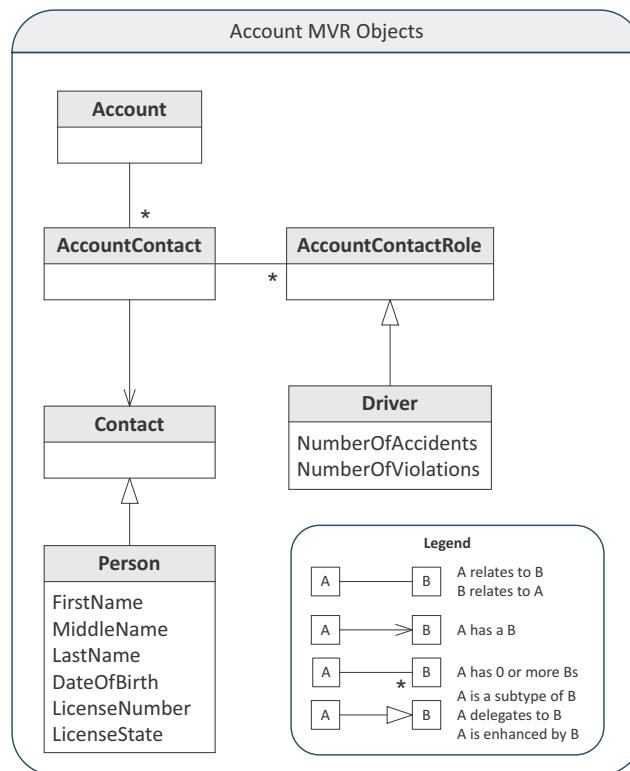
- LicenseNumber
- LicenseState
- FirstName
- LastName
- MiddleName
- DateOfBirth

The gw.plugin.motorvehiclerecord.MVRSearchCriteria class defines that fields that must match.

Account Motor Vehicle Record Object Model

MVR records exist on an account for each MVR ordered for a particular driver on a policy. The LicenseNumber, LicenseState, FirstName, MiddleName, LastName, and DateOfBirth must match the values for these fields of the MVR data at the system level.

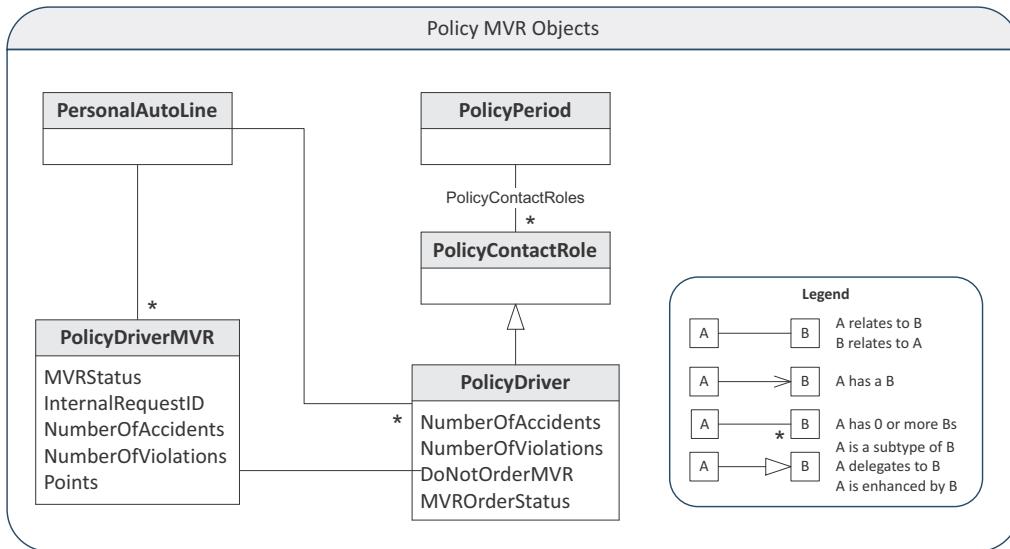
MVR data at the account level contains the latest number of accidents and violations entered on the Drivers → Driver Details → Roles tab in a policy transaction. The number of accidents and violations is stored on the Driver entity. These values are used on the next new term for that contact.



Policy Motor Vehicle Record Object Model

When an MVR is ordered, PolicyCenter creates a **PolicyDriverMVR** entity for each **PolicyDriver**. There is a way to not order an MVR (manual process). It is also defaulted to not order when a driver is excluded.

When a policy uses an MVR, PolicyCenter creates the **PolicyDriverMVR** entity.



Policy Driver

The **PolicyDriver** entity stores the number of accidents and violations entered in the **Drivers → Driver Details → Roles** tab for a policy period.

Policy Driver MVR

The **PolicyDriverMVR** entity stores the number of accidents and violations from the motor vehicle record. The workflow updates these values.

The **InternalRequestID** field on the **PolicyDriverMVR** entity matches the field of the same name on an **MVROrder** entity. This establishes the link between the policy driver MVR data and at the system MVR data in the **MVROrder** entity.

For quick display on the **Policy Drivers** screen, the **MVRStatus** field is a copy of status of the **MVROrderStatus** field on the **MVROrder** entity.

Configuring Personal Auto

This topic describes how to configure the personal auto line of business.

Configuring Copy Coverages for Personal Auto

When you select to copy coverages from a vehicle, PolicyCenter calls the `copyCoverages` method in `gw.lob.pa.PersonalVehicleEnhancement`. This method uses the `gw.coverage.AllCoverageCopier` to copy the coverages.

Configuring Personal Auto Motor Vehicle Records

This topic describes how to configure motor vehicle records in personal auto.

System Table to Set Parameters for Retrieving Personal Auto Motor Vehicle Records

When you click **Retrieve MVR** on the **Drivers** screen of a personal auto policy transaction, PolicyCenter checks to see if an MVR report already exists and is not stale. The MVR must match the account search criteria. If an MVR is found, it is retrieved. Otherwise, a new MVR is ordered. For more information about account search criteria, see “Linking Motor Vehicle Record to Account and Policy Records” on page 274.

The MVR report contains the driving record for a specified number of years from the current date.

The jurisdiction and underwriting company determine the values that PolicyCenter uses for determining whether a report is stale and how many years of driving history the report contains.

You can use the **MVRConfig** system table, **motor_vehicle_record_configs.xml**, to configure the criteria that PolicyCenter uses to search for motor vehicle records. This system table has the following columns:

Column	Description
State	The jurisdiction.
UWCompanyCode	The NAIC code for the underwriting company. The NAICCode is specified on the UWCompany tab in the underwriting_companies.xml system table.
YearsToRequest	Number of years to search backwards for a MVR. If 7, search backward 7 years from the current date.
StaleDays	The number of days after that must elapse before the motor vehicle record becomes stale. If this value is 90, then on the 90th day after obtaining an MVR report, the report is considered stale.

PolicyCenter searches for a match of the driver’s license **State** and **UWCompanyCode** of the policy to find values for **YearsToRequest** and **StaleDays**. PolicyCenter uses the first match found in the **MVRConfig** system table in the following order:

1. Both **State** and **UWCompanyCode** match
2. **State** matches and **UWCompanyCode** is not specified
3. **UWCompanyCode** matches and **State** value is not specified
4. **State** and **UWCompanyCode** are not specified

Note: If there are multiple rows with the same values for **State** and **UWCompanyCode**, PolicyCenter uses the values from the row with the first occurrence of those values.

Workflow for Personal Auto Motor Vehicle Records

When you click **Retrieve MVR** on the **Drivers** screen of a personal auto job, PolicyCenter starts the **ProcessMVRsWF** workflow. This workflow calls a plugin that retrieves an MVR report for each selected driver. In the default configuration, PolicyCenter calls a demonstration plugin, **DemoMotorVehicleRecordPlugin**, which does not call an external system. You can replace this plugin when you integrate with an external system.

When all MVR reports have been received, the workflow step **CreateActivityReceived** creates an activity with one of the following subject lines:

- **All Clear** – No accidents or violations on any MVR reports
- **Not All Clear** – at least one accident or violation on an MVR report
- **No hit** – No MVR reports found

The activity is assigned to the user who requested the MVR report.

Plugin for Personal Auto Motor Vehicle Records

In the default configuration, PolicyCenter accesses the MVR plugin through the `InternalMVRService` class. The `InternalMVRService` class implements the `IMVRService` interface.

The MVR plugin is `IMotorVehicleRecordPlugin`. In the default configuration, the plugin is the demonstration plugin, `gw.plugin.motorvehiclerecord.DemoMotorVehicleRecordPlugin`. The `IMVRService` sends requests to the plugin to order MVR reports. The `IMVRService` receives MVR reports. The demonstration plugin generates arbitrary data for each MVR report without connecting to a service provider.

Your implementation of the MVR plugin depends on your MVR provider. The `IMVRService` and `IMotorVehicleRecordPlugin` have abstract parameters for the data that is transferred between caller and the implementation of the interface. Your classes that implement these interfaces can define the parameters for data transfer.

Note: In the `DemoMotorVehiclePlugin`, to generate an MVR report that contains accidents and violations, enter a driver whose last name contains *hit*.

See also

- “Motor Vehicle Record (MVR) Plugin” on page 245 in the *Integration Guide* for information about how to integrate with a motor vehicle records provider.

Underwriting Rules for Motor Vehicle Records

In the default configuration, there is one underwriting issue type for vehicle reports. This issue type verifies that the number of accidents and violations on the policy matches the numbers on the MVR report. If the values the policy and MVR report values differ, then underwriter approval is required to bind the policy.

The underwriting issue type for motor vehicle records, `PAMVRAccidentsViolations`, is defined in the `uw_issue_types.xml` system table.

The checking set, `MVR Issues`, is evaluated at quote, quote release, bind, and issuance. The checking set is defined in `gw.job.JobProcessUWIssueEvaluator.gs`. The evaluator class, `PA_UnderwriterEvaluator.gs`, contains code that determines whether to raise an issue for this checking set.

The blocking point is at bind.

See also

- “Underwriting Authority” on page 411
- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*

Workers' Compensation

The PolicyCenter workers' compensation line of business is designed to collect data to evaluate, rate, issue, modify, and renew policies. You can combine multiple jurisdictions on a single policy. You can issue multiple policies concurrently for a single account.

The workers' compensation implementation tools adhere to North American standards such as NCCI, WCIO, and state bureaus, for determining:

- Classifications – Including multiple descriptions per code
- Exposure data
- Principal coverages – Both jurisdiction and federal
- Principal non-coverage elements – Include waivers and participating plans
- Forms and notices – Both national and jurisdiction specific

This line of business contains a reference implementation that you can use to accelerate your implementation. This line of business includes reference implementations for jobs, policy file screens, sample rating rules, sample eligibility rules, and forms logic. The reference implementation also provides sample content.

Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can build your own compliance system. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction over time. The default application contains a sample set of these classifications.

This topic includes:

- “Key Features of Workers’ Compensation” on page 280
- “Workers’ Compensation Screens” on page 283
- “Workers’ Compensation Object Models” on page 295

Key Features of Workers' Compensation

The default workers' compensation application in PolicyCenter provides many of the features you might expect in a workers' compensation policy. Key features are described in the following sections:

- “Policy Term” on page 280
- “Jurisdictions” on page 280
- “Workers’ Compensation Options” on page 282

Another key feature of workers’ compensation is audits, which includes final audit and premium reporting. For more information, see “Premium Audit Policy Transaction” on page 143.

Policy Term

In workers’ compensation policies, PolicyCenter supports an annual policy term of up to one year plus 16 days.

To view the screen that displays the policy term, see “Policy Info Screen for Workers’ Compensation” on page 284.

Jurisdictions

Workers’ compensation insurance has a wide variety of jurisdictional requirements. In the default application, PolicyCenter handles the following at the jurisdictional level: state IDs, class codes, modifiers, forms, multiple rating periods, and governing law.

To view individual screens related to jurisdictions, see “State Coverages Tab” on page 287.

State IDs

PolicyCenter supports both interstate and intrastate IDs. An intrastate ID applies to a single state or jurisdiction. You enter the intrastate ID in the details. An interstate ID, such as an NCCI Interstate ID, is shared among a group of states or jurisdictions and is entered once for the policy.

Class Codes

The default application controls available classifications by jurisdiction. If you enter data for a location in California, only class codes for California display. PolicyCenter stores workers’ compensation class codes and descriptions in the `wc_class_codes.xml` table. This table includes short and long descriptions for each class and class indicators for single classifications with multiple descriptions. You may also indicate an “if any” classification without entering a basis amount.

Multiple Rating Periods

PolicyCenter allows you to create separate rating periods for an anniversary rating date and dates selected by the user.

The anniversary rating date is a date within 12 months prior to the policy effective date. If you enter an anniversary that is not the policy effective date, PolicyCenter splits rating into two periods around the anniversary rating date.

The anniversary rating date is set at the jurisdictional level. When an anniversary rating date designates that the policy must be split, the rating period that contains that date is split into two around the anniversary rating date. For each rating period, PolicyCenter displays editable fields for covered employee information and modifiers which are specified as **Split on Anniversary** in Studio.

In addition, the user can create rating periods for each jurisdiction around one or more split dates.

Modifiers

Modifiers capture information relevant to the pricing of a policy. The rating engine uses modifiers to adjust the policy premium or some portion of the premium. Modifiers are set at the jurisdictional level and typically apply for the duration of the policy term. Some modifiers may be designated for each period if the policy has multiple rating periods.

PolicyCenter supports a wide variety of modifiers, including experience modifiers and workers' compensation scheduled credits. Various modifier types such as rate, boolean, date, and typekey are available. Modifiers may be configured to accommodate state requirements such as value ranges, required justification, or multiple rating periods.

Modifiers are defined in the policy line. For more information, see "Quote Modifiers" on page 51 in the *Product Model Guide*.

Forms

The workers' compensation application obtains detailed information which PolicyCenter uses to infer forms and to complete the data used on the forms. An individual form can be identified as applying to all jurisdictions or to specific jurisdictions. You can integrate PolicyCenter with your form engine. PolicyCenter allows you to view forms in the user interface and to integrate with form creation and printing systems. For more information, see "Policy Forms" on page 461.

Governing Law

Most policies designate basis amounts for workers' compensation act classes. PolicyCenter also accommodates designating other governing laws for covered employee exposures. The governing laws in the default application are:

- **State Act** – Default. Coverage under normal workers' compensation laws in a jurisdiction.
- **Voluntary Comp** – Extension of jurisdictional law to offer coverage to a class not required by law, such as domestic or farm workers.
- **U.S.L.&H.** – The United States Longshore & Harbor Workers' Compensation Act provides coverage for work performed adjacent to navigable waterways. Uses jurisdictional class codes but with different rates and benefits.
- **Outer Continental Shelf Act** – Coverage for work performed in coastal waters, such as offshore oil rigs. Uses jurisdictional class codes but with different rates and benefits.
- **Fed Coal Mine Act** – Coverage for coal mine workers. Uses jurisdictional class codes but with different rates and benefits.
- **Migrant & Seasonal Agricultural Workers Act** – Coverage for migrant farm workers. Uses jurisdictional class codes but with different rates and benefits.
- **Defense Base Act** – Coverage for U.S. based employees working on military bases both domestically and in foreign countries. Uses jurisdictional class codes but with different rates and benefits.
- **Non-appropriated Fund Instrumentality's Act** – Coverage for U.S. based employees working in military PXs both domestically and in foreign countries. Uses jurisdictional class codes but with different rates and benefits.
- **Limited Maritime** – Creates an underwriting flag rather than offering distinct coverage or benefits. It typically indicates that an employee is proximate to an Admiralty/Maritime/Jones Act exposure but is covered under workers' compensation defined benefits.
- **Exposure Related Stop Gap** – In a monopolistic jurisdiction, insurance companies can write Stop Gap coverage for employer's liability insurance.

You can enter the class code and basis amount for each governing law in the covered employees section. A single class code may have multiple descriptions. These descriptions are important in printing policies and audits and for class code search. An example of a class code with multiple descriptions is code 8742 for the jurisdiction of

California. For information about configuring multiple descriptions, see “System Tables” on page 75 in the *Product Model Guide*.

Governing laws affect rates, forms inference, and benefits paid to an injured worker. Although many of these refer to federal acts, all the governing laws refer to workers’ compensation type programs that provide defined benefits. Do not confuse these with federal liability acts such as FELA and Maritime, which are described in “Specialty Operations” on page 283.

Workers’ Compensation Options

In the default application, PolicyCenter handles a number of workers’ compensation options such as inclusions and exclusions, policy plan type, and speciality operations.

To view individual screens related to options, see “WC Options Screen for Worker’s Compensation” on page 289.

Inclusions and Exclusions

PolicyCenter allows the following inclusions or exclusions from the policy:

- **Include or exclude individuals** – In general, jurisdictions require employees to be covered by workers’ compensation insurance. There are a variety of circumstances in which employee status requires clarification. As a result of these clarifications, individuals or classes of individuals may be included or excluded from coverage. All entries have significant impact on coverage and forms inference.
- **Include or exclude owners and officers** – The application allows you to specify owners and officers with their jurisdiction, title, and ownership percentage. Each officer may be designated as included or excluded. You must enter a classification for included officers.
- **Exclude operations or jobs** – The exclusions option allows the insured company to exclude operations or jobs. For example, the insured may want to exclude a construction site because another type of policy covers this site.

Policy Plan Types

PolicyCenter includes the ability to designate and define an appropriate policy plan such as a retrospective rating or participating plan. PolicyCenter stores the data for forms but does not do any calculations for retrospective rating or participating plan.

Retrospective Rating Plan

A retrospective rating plan looks at the loss experience of the policy after its expiration in order to recalculate the premium. The policy is initially rated and issued with the appropriate jurisdictional rates. At predetermined intervals after policy expiration, the premium is recalculated based on agreed upon factors, and the amount paid by the policyholder for the policy term is adjusted.

PolicyCenter stores the values needed to administer the plan and calculations; you will need to implement the premium calculations including the linking of multiple concurrent policies.

Participating Plan

A participating plan looks at the experience of all policyholders participating in the plan to determine profitability of the plan sometime after policy expiration. This experience may result in a dividend to the plan’s policyholders. Although there is considerable variability in plan design, some common elements are:

- Plan ID – Defines all factors which are invariable for that specific plan.
- Retention – The percent of the premium that the carrier always keeps.
- Loss Conversion Factor – A factor which is applied to losses when calculating dividends.

Specialty Operations

PolicyCenter supports specialty operations such as federal liability, waivers of subrogation, employee leasing, and aircraft seat surcharge.

Federal Liability

This coverage departs from workers' compensation principally in two ways. First, it is tort based rather than being a no fault defined benefits system. Second, it applies to only two industries: the operation of US flag vessels and the operation of railroads. Maritime coverage is also referred to as Admiralty or Jones Act. Liability for railroad operations is typically referred to by the acronym FELA.

The federally sanctioned programs are:

- **Program I** – This program is pure tort. Recovery is based on determining fault.
- **Program II** – This program gives the injured employee the option of tort relief or benefits under workers' compensation jurisdictional act or U.S.L.&H.

After selecting a program and selecting the appropriate federal liability law, you can enter federal liability class codes for different types of employee activity. For Program II, these class codes subsequently translate into domain specific jurisdiction or U.S.L.&H. codes. The user interface does not display the domain specific codes. The federal liability class codes appear in the printed policy. All codes impact coverage and forms inference.

Waivers of Subrogation

This is a contractual agreement between the insured and the carrier to prevent the carrier from subrogating to a named third party in the event of a loss. A waiver can be on a blanket basis which applies to all workers' compensation exposures. A waiver can also be on a specific basis which applies to a named job, contract, or event. In the case of specific waivers, exposure information is collected to calculate a charge for the job, contract, or event. A carrier may choose to specify a flat charge for these waivers or waive specific charges (as a matter of policy or on an exception basis) until final audit.

Employee Leasing

The employee leasing option allows you to define contractual information if the named insured is either a labor contractor or obtains employees from such a contractor. This information includes names and dates for contracts and whether the policy includes or excludes coverage. If the employer is a labor contractor and supplies employees to others, you can specify labor clients. If the employer obtains employees from others, the details are about that labor contract.

Aircraft Seat Charge

The aircraft seat charge option allows you to obtain details required for a passenger seat surcharge and endorsement per aircraft.

Workers' Compensation Screens

The workers' compensation implementation contains a series of screens to pre-qualify the applicant, describe locations, choose coverages, assess risk, quote the policy, and select payment options. This section provides descriptions of fields to define the policy contract and to obtain related information.

- “Qualification Screen for Workers’ Compensation” on page 284
- “Policy Info Screen for Workers’ Compensation” on page 284
- “Locations Screen for Workers’ Compensation” on page 286
- “WC Coverages Screen for Workers’ Compensation” on page 286
- “Supplemental Screen for Workers’ Compensation” on page 289

- “WC Options Screen for Worker’s Compensation” on page 289
- “Risk Analysis Screen for Workers’ Compensation” on page 291
- “Policy Review Screen for Workers’ Compensation” on page 291
- “Quote Screen for Workers’ Compensation” on page 292
- “Forms Screen for Workers’ Compensation” on page 292
- “Payment Screen for Workers’ Compensation” on page 293

Qualification Screen for Workers’ Compensation

This screen contains a set of questions to pre-qualify the applicant.

The questions reflect risks that the carrier wants to assess at the beginning of the submission. Since this is the gateway to the balance of the submission, the intent is to determine eligibility of the applicant.

The questions do not contain any regulatory requirements. In your implementation, the question set can impact later functionality and contain regulatory requirements.

In the default implementation, some of the pre-qualification questions raise underwriting issues if the answer is not the correct answer. In Studio you can specify the correct answer and whether or not to raise an underwriting issue.

See also

- “Underwriting Authority” on page 411
- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*

Policy Info Screen for Workers’ Compensation

The **Policy Info** screen captures basic information about the policy.

This screen contains information about the primary named insured, producer information, policy details, and selected underwriting company. The buttons and choices that you see on the **Policy Info** screen vary based on the contacts associated with the account and how they are associated with the policy. For example, under **Primary Named Insured** you might see any or all of the following:

- New Company
- New Person
- From Address Book
- Existing Contact

Not all choices appear at any given time.

The following table describes key fields of the **Policy Info** screen.

Field name	Description
Primary Named Insured	<p>PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select:</p> <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact <p>If the named insured is a person, the social security number is required in Official IDs. If the named insured is a business, then FEIN is required.</p> <p>For Existing Contact, PolicyCenter lists contacts with the Account Holder or Named Insured role on the account. The list does not include contacts who are already the primary or additional named insureds on this policy.</p>
Business and Operations	<p>Designate if the policy is assigned risk. For Assigned Risk the default value is No. Assigned Risk may be set to Yes if a carrier directly writes or services this market segment and the submission qualifies as an assigned risk.</p> <p>If a carrier does not service or write directly in the non-standard market, change the configuration so that this element is hidden.</p> <p>The Organization Type drop-down is required. Some of the values on this drop-down are: Common ownership, Corporation private, Corporation public, Individual. This value affects forms inference.</p>
Organization Type	<p>Enables you to select the type of organization such as whether this business is a corporation or partnership.</p>
Additional Named Insureds	<p>Use to create additional named insureds. This might be a business partner, for example. The Add button enables you to select:</p> <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Additional Named Insured
Policy Details	<p>When Term Type is set to Annual, it has an editable Expiration Date field. This allows you to set an annual policy to one year plus 16 days. Use Other to specify short term periods.</p> <p>The term types are defined in Configuration → Product Model → Products → Workers' Compensation in Studio.</p>
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p> <p>If you change the producer in the middle of a policy period, the new producer is the Producer of Service. The original Producer of Record remains. When the policy is renewed, the Producer of Service becomes the Producer of Record.</p>
Underwriting Companies	<p>The underwriting company is automatically set for a policy version based on logic defined in segmentation classes. You can select a different underwriting company if you have the correct permissions.</p> <p>The drop-down list contains a configurable set of choices.</p>
Preferred Currency	<p>If PolicyCenter is configured as a multicurrency system, the Policy Info screen displays this label and the following fields related to the preferred currency.</p>

Field name	Description
Coverage	The preferred or default currency for coverages on the policy. The Coverage currency choices come from the policy line configuration in Product Designer. In the base configuration, the default is the preferred coverage currency on the account, Account.PreferredCoverageCurrency.
Settlement	The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the Quote and other screens. In the base configuration you can select one of the following currencies: <ul style="list-style-type: none">• USD• EUR• GBP• CAD• AUD• RUB• JPY The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, Account.PreferredSettlementCurrency. You can set the preferred settlement currency in the Account File Summary screen Currencies → Settlement field.

Locations Screen for Workers' Compensation

In the Locations screen, enter the locations that this policy covers.

By default, the primary location defaults to the primary location on the account. Add other locations that are covered by this policy. If you create a new location, PolicyCenter adds it to the account. Select a location and click Set As Primary to change the primary location.

Workers' compensation locations include additional fields intended to meet requirements for proof of coverage (POC) filings. You have the option to create a non-specific location if a particular address within an jurisdiction is unknown.

Field name	Description
Employer (if other than Primary Named Insured)	The drop-down list displays the Named Insureds from the Policy Info screen.
Number of employees at this location	This field satisfies a POC requirement.
SIC code (if different from primary code)	This field satisfies a POC requirement. You can enter a location SIC code when it varies from the IndustryCode recorded on the Account. You can use this code for filtering or validation.

For more information about locations, see “Locations” on page 343.

WC Coverages Screen for Workers' Compensation

The WC Coverages screen has State Coverages and Policy Coverages tabs where you enter workers' compensation basis and coverage information.

- “State Coverages Tab” on page 287
- “Policy Coverages and Exclusions Tab” on page 288
- “Class Code Search Screen” on page 289

State Coverages Tab

This tab displays the jurisdictions covered on the policy. Clicking the jurisdiction in the listview entry opens the appropriate **State Details** pane. This pane contains all the detail required for covered employees. It also includes available modifiers, deductibles and official IDs for the jurisdictions listed.

Rating Periods

PolicyCenter allows you to create multiple rating periods for a jurisdiction in two ways. The rating period can be split around an anniversary rating date. The rating period can be split around one or more split dates specified by the user.

The **Anniversary Date** defaults to the policy effective date. You can set the anniversary rating date to a date within 12 months prior to the policy effective date. If you enter an anniversary that is not the policy effective date, PolicyCenter splits rating into two periods around the anniversary rating date. For example, a policy has an effective date of 01/01/2010. You set the **Anniversary Date** to 07/01/2009. PolicyCenter creates two rating periods: one from 01/01/2010 to 07/01/2010 and another from 07/01/2010 to 01/01/2011.

The rating period for each jurisdiction can be split around a user-defined date split date. Select the jurisdiction and click the **Split Period** button. Specify the **Split Date** and the **Type**. In the default configuration, the choices for **Type** are **Forced Rerating** and **Late Modifier**.

When you return from the **Split Period** screen, click the **Update All Basis** button, or leave this page, PolicyCenter splits the jurisdiction-specific deductible, class values and certain modifiers. PolicyCenter splits the class values and these modifiers into two periods: one before and one after the split date. Policy term basis amounts are prorated by default and are editable.

On renewal, the anniversary rating date is reset to the renewal effective date but remains editable. The user-defined split periods are also removed.

For more information about anniversary rating dates and modifiers, see “Quote Modifiers” on page 51 in the *Product Model Guide*.

Workers’ Compensation State-Specific Deductible

You can configure jurisdiction-specific deductibles. If a deductible is available and selected, the application displays the appropriate deductible options.

State IDs

This section allows you to enter IDs for states or jurisdictions. You can validate the format and specify whether the ID is required. For configuration information, see “Defining Official IDs” on page 39 in the *Product Model Guide*. You can view the ID formats in `official_id_validation_info.xml` by navigating to **configuration** → **System Tables** in Studio.

Modifiers

The modifiers section displays all modifiers for the current jurisdiction that are effective for the duration of the policy period. When a jurisdiction has multiple rating periods, modifiers that are set to **Split on Anniversary** appear once for each period.

- Modifiers may be defined as boolean, date, rate, or typekey. A boolean modifier indicates whether the policy is eligible and depends on the rating engine to apply the appropriate factor. The other modifier types have a variable value, and PolicyCenter passes the value to the rating engine.
- Minimum/maximum ranges may be applied to numeric modifiers.

The **WC Schedule Credits** are modifiers specific to workers’ compensation. Click the numbered link next to **Schedule Credits** on the **WC Coverages** screen to display the **WC Scheduled Credits** screen shown below.

The **WC Schedule Credit** worksheet appears for jurisdictions which permit this credit. There are minimum and maximum values for the credit overall (typically limited by regulation to +/- 0.25), with separate minimum and

maximum values per category. PolicyCenter passes the **Overall** value to the rating engine; the category values and justifications must be preserved for regulatory and internal reviews.

You can view this screen in Studio by navigating to `WCScheduleCreditPopup.pcf`.

For more information about modifiers, see “Quote Modifiers” on page 51 in the *Product Model Guide*.

Covered Employees

Add or remove classes of employees covered at the location. Enter a **Basis** amount for the policy period. Use a separate entry for each governing law, location, and class code combination.

When multiple rating periods are created around an anniversary rating date or split date, the class codes are split into those periods. Payroll amounts are divided on a pro rata basis based on the rating period dates. Any covered employee basis entered before the rating period split is split pro rata but remains editable.

Field name	Description
Governing Law	You can enter classification and basis amounts for each governing law under the jurisdiction. State Act is the default governing law for each classification. For more information on Governing Law options, see “Governing Law” on page 281. These options are configured in the SpecialCov typelist.
Location	The Location drop-down lists locations that have already been entered in the Locations screen for the current jurisdiction. If you need to add an exposure to a location not in the list, you need to go back to the Location screen to create that location.
Class code	After selecting a location, the Class Code allows you to enter a code or search for a code. The default application contains definitions for many jurisdiction and NCCI class codes. For more information on searching for class codes, see “Class Code Search Screen” on page 289.
Description	This read-only field displays the description of the class code.
# Employees	This field is not required for rating, but is available for such things as POC reporting, catastrophe analysis, and wage level analysis.
If any	Selecting this field disables the Basis field. Select this field if you have a class that may not have any basis this year.
Basis	This field is required field unless you select If any . Enter the exposure or payroll amount for this class. If you set a midterm anniversary rating date, clicking on the Update All Basis button in the toolbar generates two rows for payroll entry and two sets of effective dates. These split around the anniversary rating date. The split automatically prorates payroll across the periods. These values may be edited to reflect seasonality or other business requirements.

Policy Coverages and Exclusions Tab

The **Policy Coverages and Exclusions** tab allows you to select the policy level coverages. Required coverages for the jurisdiction cannot be unselected.

The following table describes some of the fields in the **Policy Coverages and Exclusions** tab. Other fields may appear depending upon the covered jurisdictions.

Field name	Description
Workers' Compensation States	Displays the covered jurisdictions. These are the jurisdictions with covered employees.
Workers' Comp Employer's Liability → Employer's Liability Limit	Identify the limit coverage term for employer's liability coverage. The drop-down list displays package or multi-part limits as configured in Studio. The limits are defined: <ul style="list-style-type: none"> • Per accident • Disease per employee • Disease per policy

Field name	Description
Workers' Comp Employer's Liability → Stop Gap	Specify the monopolistic jurisdictions covered by employer's liability Stop Gap coverage. The options are: <ul style="list-style-type: none"> • All monopoly states • Listed states only • None
Other States Insurance → Covered States	The Covered States allows you to select jurisdictions that conditionally require workers' compensation insurance. For example, the applicant may have no permanent operations or locations in these jurisdictions, but employees may be there on temporary assignment. The choices are: <ul style="list-style-type: none"> • All other non-monopolistic states – Indicates all non-covered jurisdictions that are non-monopolistic. • All states except – Enter a comma separated list of USPS jurisdiction codes. Invalid entry data is the jurisdictional code for any monopolistic jurisdiction or any covered jurisdiction. • Listed states only – Enter a comma separated list of USPS jurisdiction codes. Invalid entry data is the jurisdiction code for any monopolistic state and/or any covered jurisdiction. • None – No temporary employee exposure anticipated outside of jurisdictions with permanent facilities. Employees of other jurisdictions are not covered unless specifically added to the policy by endorsement.
Statutory Workers' Comp	Indicates workers' compensation coverage as required by the covered jurisdictions. No additional coverage terms apply to this coverage.
Exclude Medical Option (Hospital Only)	This option is available only for hospitals and provides workers' compensation without medical benefits. This is different from medical deductibles or medical reimbursement plans.

You can view this tab in Studio by navigating to `WorkersCompCoverageCV.pcf`.

You can configure policy coverages in the **Coverages** page in Product Designer under **Product Model → Policy Lines → Workers' Comp Line**. You can configure or add coverages such as Workers' Comp Employer's Liability and set available terms and their options such as liability limit choices. For more information, see “Working with Policy Lines” on page 23 in the *Product Model Guide*.

Class Code Search Screen

You can search for class codes on the **Class Code Search** screen. All search fields are optional. You can enter the following:

- **Code** – Enter the class code number. PolicyCenter searches for a code that starts with the value you enter.
- **Classification** – Enter the classification that you are searching for. PolicyCenter searches for classifications that contain the value you enter. The search is not case-sensitive.

Supplemental Screen for Workers' Compensation

This screen contains a question set in the middle of the submission process. This page can be enhanced to present different question sets based on class codes or covered jurisdictions. Questions and answers can also be used to calculate scheduled credits or debits.

You can view this screen in Studio by navigating to `WorkersCompSupplementalScreen.pcf`. The screen displays the question set defined in **Product Model → Question Sets → WCSupplemental** in Product Designer.

WC Options Screen for Worker's Compensation

While workers' compensation has relatively few coverages, it has many contractual provisions with extended information needs. The **WC Options** screen enables you to select the options required for a given policy. When you select an option, a popup screen appears that allows you to enter information for that option.

You can view this screen in Studio by navigating to `CoverageOptionsScreen.pcf`.

The following sections describe each option:

- “Federal Liability” on page 290
- “Waivers of Subrogation” on page 290
- “Owners/Officers” on page 290
- “Individuals Included/Excluded” on page 290
- “Retrospective Rating Plan” on page 290
- “Participating Plan” on page 290
- “Employee Leasing” on page 291
- “Aircraft Seat Change” on page 291
- “Exclusions” on page 291

You can view these panels in Studio by navigating to `WCOptionsPanelSet.pcf`.

Federal Liability

This option is available in two federally sanctioned programs: **Program I** and **Program II**. If you select **Program II**, you can enter federal liability class codes for different types of employee activity; these class codes subsequently translate into domain specific jurisdiction or U.S.L.&H. codes. The user interface does not display the domain specific codes. The federal liability class codes appear in the printed policy. All codes impact coverage and forms inference.

You can view these class codes in the `workers_comp_federal_liability_class_codes.xml` system table in Product Designer.

Waivers of Subrogation

Multiple waivers of either blanket or specific type may be defined on this screen. There is built in validation for specific waivers that filter for jurisdiction and class code combinations that appear on the **State Coverages** tab. For example, the sum of the **Project Payroll** cannot exceed the basis on the **State Coverages** tab for that jurisdiction and class code. See “**State Coverages Tab**” on page 287.

Owners/Officers

You may enter owners and officers of the named insureds on this screen. Inclusion/exclusion form inference uses this information. Each listed person may be designated as included or excluded from the policy. For those included, you must specify the jurisdictional classification which includes remuneration of the covered person.

Individuals Included/Excluded

Specify included or excluded persons on this screen. All entries have significant impact on coverage and forms inference.

Retrospective Rating Plan

When retrospective rating applies to a policy, the default application stores the retrospective rating requirements on forms and in rating calculations.

Participating Plan

You can designate a carrier participating plan on this screen. The **Participating Plan** tab displays the following fields:

- **Plan ID** – Defines all factors which are invariable for that plan.
- **Retention** – The percent of the premium that the carrier always keeps.
- **Loss Conversion Factor** – Enter a factor which is applied to losses when calculating dividends.

The default application stores participating plan requirements but does not include these requirements on forms or use them for calculating dividends. The default implementation supports a single dividend plan per policy.

Employee Leasing

You can define labor contracts on this screen. The data that is required for this option depends upon the following:

- Do you supply or receive employees? If you supply employees, add a **Client**. If you receive employees, add a **Supplier**.
- Does this policy include or exclude those employees? Select this on the **Contact Detail** screen.

All entries have significant impact on coverage and forms inference.

Aircraft Seat Change

The **Aircraft Seat Change** option allows the insured company to define aircraft used by company as required by manual rules. PolicyCenter uses this information for rating and forms inference.

Exclusions

The **Exclusions** option screen allows you to enter excluded groups. For example, you can exclude employees working at a job contract site that a separate workers' compensation policy covers.

Manuscript Option

The **Manuscript Option** screen allows you to enter the manuscript text for a custom coverage, exclusion, or policy condition. You can also specify a premium amount.

Risk Analysis Screen for Workers' Compensation

The **Risk Analysis** screen allows you to enter data that is used to evaluate the risk of the applicant.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – This tab appears when viewing a policy. It does not appear in a work order such as a submission. This tab allows you to add an underwriting referral reason to a policy. Underwriting referral reasons are usually only used when a notable condition arises outside of a job, possibly outside of the data that PolicyCenter maintains on the policy. For more information about this tab, see “Working with Underwriting Referral Reasons” on page 416.
- **UW Issues** – This tab displays underwriting issues that were raised during the job. For more information on this tab, see “Working with Underwriting Issues” on page 414.
- **Prior Policies** – This tab displays information about prior policies usually with another carrier. You can add information about prior policies.
- **Claims** – This tab allows you to search for loss claims on a related policy or by loss date. When integrated with a claim system, this tab displays claims from the claim system. For more information, see “Viewing Loss Claims for Policies” on page 716.
- **Prior Losses** – This tab displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

Policy Review Screen for Workers' Compensation

The **Policy Review** screen is similar to the summary screen for other lines of business.

For a submission policy transaction, this screen contains all the policy data in summary form and is similar to the summary screen for other lines of business. For other policy transactions, this screen displays the differences

between the policy versions. It is useful to review this screen before generating a quote. If any of the information needs to be changed, click the menu link on the left sidebar to edit the appropriate screen.

Quote Screen for Workers' Compensation

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen.

This screen is similar to the quote screen for other lines of business. It contains the basic information at the top including the total premium and taxes and surcharges. The bottom of the screen displays a breakdown of the premium.

Within the **Quote** screen, your choices at this point include:

- Editing the policy (and thereby invalidating the current quote)
- Creating a new version of the policy
- Saving a draft of the policy
- Binding or issuing the policy
- Withdrawing, declining, or not taking the policy
- Printing the policy

PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

In addition, Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

If you have Guidewire Rating Management, the **Policy Premium** tab has a **Show Rate Worksheet** button. For more information about this button, see “Rating Worksheets” on page 548 in the *Application Guide*.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Policy Premium Tab

The **Override Rating** button allows you to manually override the premium that the rating engine automatically generates for a policy. For more information, see “Rating Overrides” on page 445.

Forms Screen for Workers' Compensation

The **Forms** screen allows you to associate forms with a policy.

PolicyCenter does not store the content of any forms associated with a policy. Therefore, the **Forms** screen only identifies that zero, one, or more, forms have been associated with the policy. The **Forms** screen simply lists form instances that indicate the physical forms, which are usually printed by an issuance system. Each form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form. The form content is not stored within PolicyCenter.

When you click **Quote** or **Bind**, PolicyCenter infers which forms the policy needs. In the default application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy Forms” on page 461

- “Policy Form Pattern Administration” on page 683 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment Screen for Workers’ Compensation

The **Payment** screen displays payment information for the policy.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- **Total Premium**
- **Taxes and fees**
- **Total Cost** – The sum of the previous fields.

Billing

Under the **Billing** heading, you can select:

- **Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- **Alt Billing Account** – Select an alternate billing account.
- **Billing Contact** – Select a person or company as the billing contact.

Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- **List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer’s mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample billing accounts. If you are integrated with **BillingCenter**, the plugin returns a list of billing accounts retrieved from **BillingCenter**.

- **Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent's account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle's account as the alternate billing account. The uncle's account is not a subaccount of the parent's account.

You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.

- **Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample installments plans. If you are integrated with **BillingCenter**, the plugin returns a list of valid installment plans for the selected list bill payer. The integration retrieves installments plans from **BillingCenter**.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the **StandAloneBillingSystemPlugin** or the default integration with **BillingCenter**, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The **StandAloneBillingSystemPlugin** simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If

you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payments system, “Payment Integration” on page 553 in the *Integration Guide*.

If you select **Automatic** for **Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.
- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

Note: The BillingCenter integration does not support this functionality.

For **Payment Method**, you can select **Installments** or **Reporting Plan**. If you select **Reporting Plan**, you must select a **Premium Reporting Plan**.

In the **Audit** section, specify whether the policy **Requires final audit**. Your choices are: **Determined By Business Rule**, **Yes**, or **No**. If you selected **Reporting Plan** as the payment method, final audit is required.

Workers’ Compensation Object Models

This section describes the objects or entities associated with the workers’ compensation line of business.

PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.

The following sections provide further details about key entities of workers’ compensation:

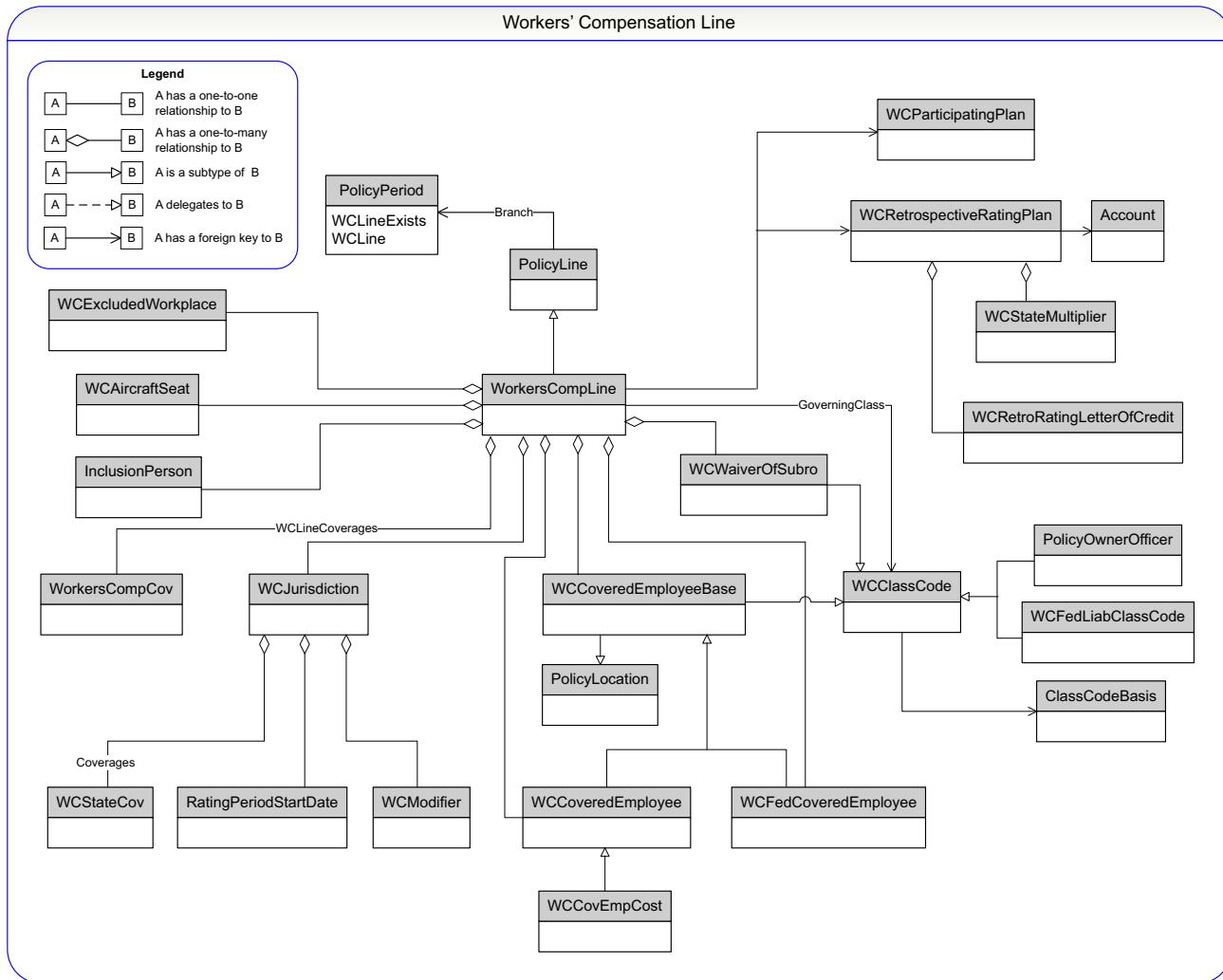
- “Workers’ Compensation Object Model Overview” on page 296
- “Policy Contact” on page 297
- “Coverages” on page 297
- “Modifiers” on page 298
- “Employees” on page 298
- “Jurisdictions” on page 299
- “Retrospective Rating Plan” on page 299

See also

- “Core Entities Associated with Policies” on page 309
- “Cost and Transaction Model for Workers’ Compensation Line” on page 433

Workers' Compensation Object Model Overview

The following diagrams show the main entities of the workers' compensation business line and relationships between these entities.



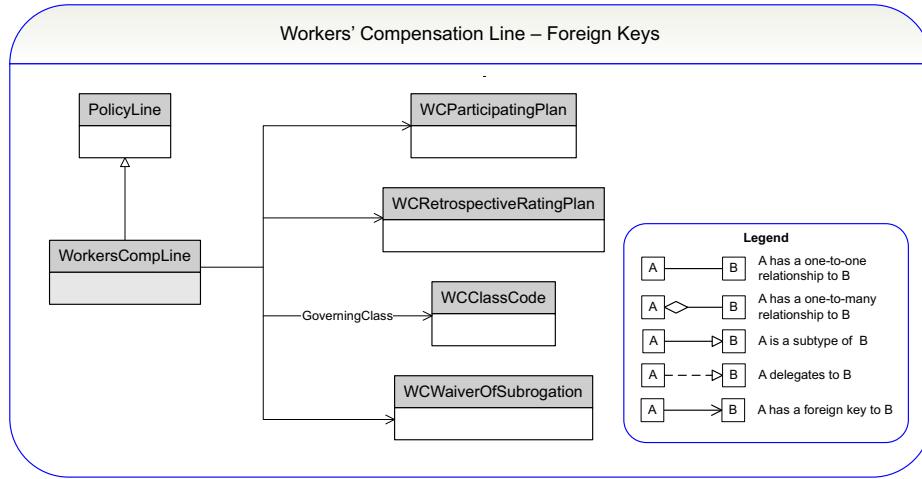
One-to-Many Relationships

The **WorkersCompLine** entity has a one-to-many relationship to:

- **WCCoveredEmployeeBase**
- **WCCoveredEmployee**
- **WCFedCoveredEmployee**
- **WCWaiverOfSubrogation**
- **WorkersCompCov**
- **WCJurisdiction**
- **WCExcludedWorkplace**
- **WCAircraftSeat**
- **InclusionPerson**

Foreign Keys

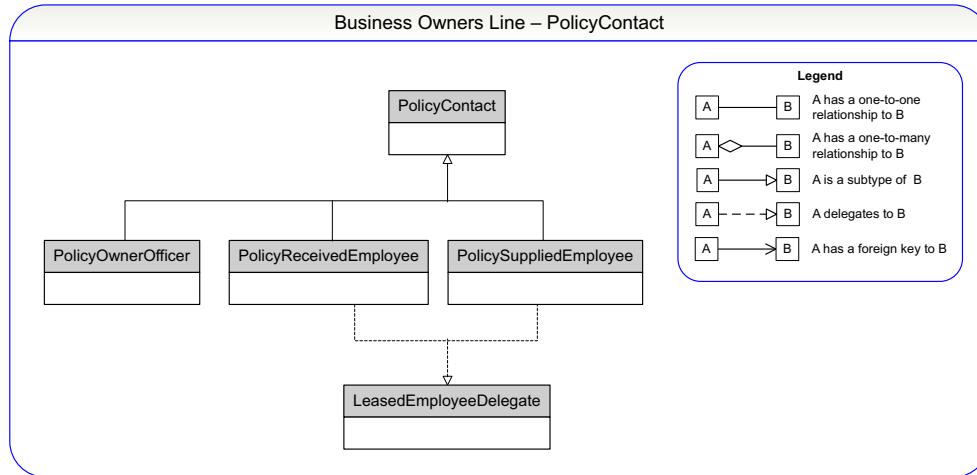
The following diagram shows some of the WorkersCompLine foreign keys.



Note: This diagram shows a partial listing of entities in the workers' compensation line. For the complete list of entities and properties, see the *Data Dictionary*.

Policy Contact

The **PolicyContact** entity can be a person or a business, and contains contact information such as name and address. The **PolicyOwnerOfficer**, **PolicyReceivedEmployee** and **PolicySuppliedEmployee** subtypes are specifically related to workers' compensation. The **PolicyOwnerOfficer** entity is for an owner or officer of the company. The **PolicyReceivedEmployee** entity contains information about employees that the insured receives from another company or person. The **PolicySuppliedEmployee** entity contains information about employees that the insured supplies to another person or company.



Note: This diagram shows a partial listing of entities in the workers' compensation line. For the complete list of entities and properties, see the *Data Dictionary*.

Coverages

PolicyCenter defines a coverage as a protection from a specific risk. A coverage entity must implement the **Coverage** interface. Coverages always attach to a coverable. While there are two types of coverages: property and liability, workers' compensation has only liability coverage. For example, on a workers' compensation policy, a liability coverage protects the worker for injury received on the job.

In the base configuration, the workers' compensation policy line contains the following types of coverages:

Coverage type	Attaches to	Description
WCStateCov	WCJurisdiction	Coverage choices for a particular jurisdiction.
WorkersCompCov	WorkerCompLine	Coverage choices that are not jurisdiction-based.

Modifiers

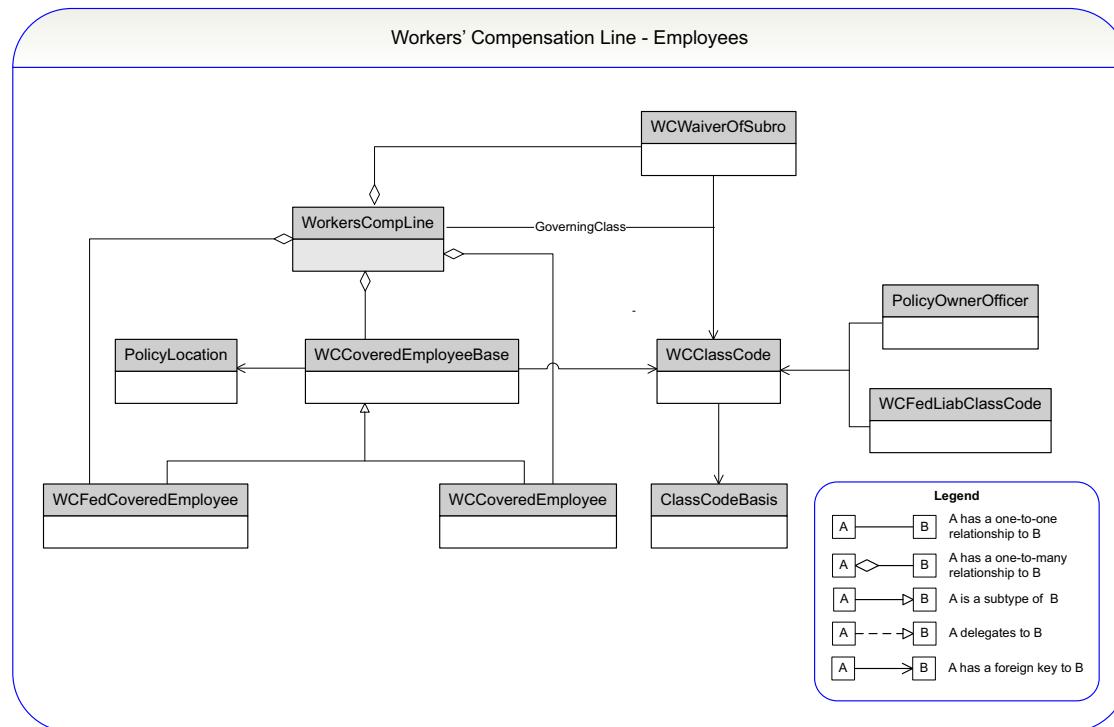
A modifier is a value used by the rating engine to adjust the policy premium or some portion of the premium. Modifiers capture information relevant to the pricing of a policy that are not necessarily tied to a specific coverable or coverage. In workers' compensation, there is the following modifier type:

Modifier type	Applies to	Description
WCModifier	WCJurisdiction	A modifier on the jurisdiction. Multi-policy discount or no-loss discount.

To learn more about how costs work in the workers' compensation line, see “Quoting and Rating” on page 419.

Employees

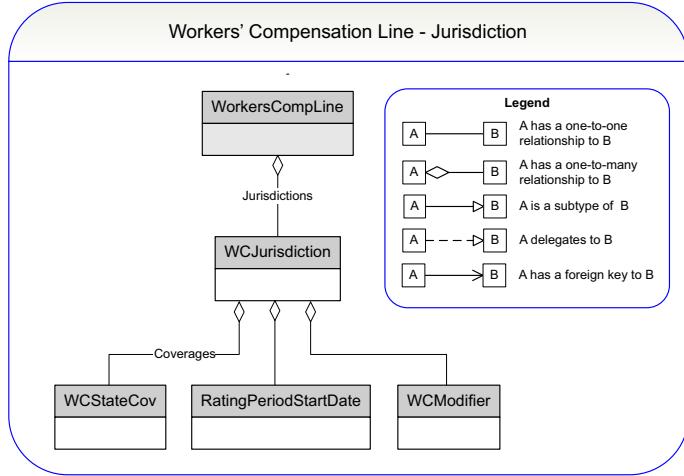
The following diagram shows entities related to employees. PolicyCenter represents each employee class by a WCCoveredEmployeeBase, a WCCoveredEmployee, or a WCFedCoveredEmployee entity. The WCCoveredEmployee subtype has an array key that allows you to get to WCovEmpCost. The WCFedCoveredEmployee subtype allows you to add a RailroadOrVessel name for Federal Liability Program I.



Note: This diagram shows a partial listing of entities in the workers' compensation line. For the complete list of entities and properties, see the *Data Dictionary*.

Jurisdictions

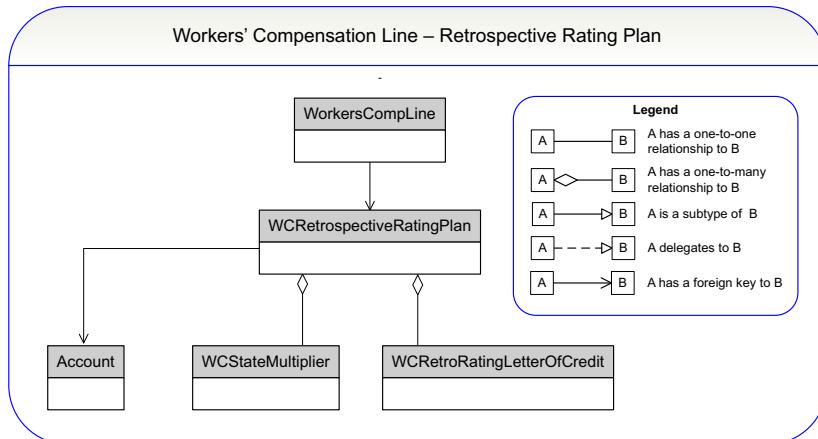
The WCJurisdiction entity maps to a covered jurisdiction. The State property is a type key to the jurisdiction. Other properties store the effective and expiration dates and, if the period is sliced, the slice date. The WCJurisdiction entity allows you to access coverages and rating information associated with the jurisdiction.



Note: This diagram shows a partial listing of entities in the workers' compensation line. For the complete list of entities and properties, see the *Data Dictionary*.

Retrospective Rating Plan

The WCRetrospectiveRatingPlan entity is used for rating the policy.



Note: This diagram shows a partial listing of entities in the workers' compensation line. For the complete list of entities and properties, see the *Data Dictionary*.

Additional Features of PolicyCenter

Policy File

In PolicyCenter, you can work on policies within the **Policy** tab. You use policy transactions to work with the policy file. This topic describes the user interface and menu actions of the policy file.

This topic includes:

- “Policy File Overview” on page 303
- “Policy File Screens, Menus, and Actions” on page 306
- “Working with Policies” on page 308
- “Configuring Policies” on page 309
- “Configuring Copy Data for a Line of Business” on page 313
- “Configuring Split and Spin-off Policies” on page 314

See also

- “Policy Transactions” on page 73

Policy File Overview

The policy file is the electronic file in which PolicyCenter stores policy information that is part of the legal contract. You use policy transactions to work with the policy file. For example, policy transactions allow you to create, modify, cancel, and perform other actions on policies.

For more information about policy transactions and how to work with the policy file, see “Policy Transactions” on page 73.

Overview of Copying Data Between Policies

PolicyCenter allows you to copy data from an existing policy or policy transaction to an open policy transaction. In the base configuration, this functionality is available in the personal auto line of business. By configuring PolicyCenter, you can add this functionality to other lines of business.

In personal auto, you may want to copy data between policies for one of the following reasons:

- An agent added a car to the wrong personal auto policy. The agent can copy the car over to the correct policy without needing to reenter the information.
- A family has a personal auto policy that covers several vehicles. A daughter moves out of the house, and her parents give her one of the vehicles. The daughter gets her own account and personal auto policy. The agent copies the vehicle from the parents' policy to the daughter's policy. The agent does not need to reenter the vehicle information in the daughter's policy.
- The daughter buys a new vehicle and returns the old vehicle to her parents. The agent reinstates the old vehicle by copying the vehicle from an earlier version of the parents' or daughter's policy onto a policy change for the parents' policy.

Searching for Policies and Policy Transactions from Which to Copy Data

You can copy data if you are in one of the following types of policy transactions:

- Submission
- Policy change
- Rewrite
- Rewrite new account
- Renewal

From within one of these policy transactions, you can search for both bound policies and policy transactions to copy data from. The search finds bound policies and policy transactions with the same product type as the current policy transaction. You can copy data from other policy terms or other policy transactions on the current policy.

When searching for policies or policy transactions to copy from, each search result represents a slice of the policy at a particular time. The slice contains the entities available from that slice of the policy.

For policy transactions, PolicyCenter displays the slice on the edit effective date of the policy transaction. For policy terms, PolicyCenter displays the last slice of the policy period.

When you select a policy transaction to copy from, PolicyCenter displays the slice on the edit effective date of the policy transaction. For policy terms, PolicyCenter displays the last slice of the policy period. For policy terms, you have an option to specify a date which represents the slice of the policy at that particular time. PolicyCenter displays the entities available from that slice of the policy.

Copy Data and Multi-version Quoting Policy Transactions

You can copy data to and from multi-version quoting policy transactions.

- **Copy data to** – When working in a multi-version quoting policy transaction, you can copy data to any version of that multi-version quoting policy transaction. PolicyCenter copies the data to the version that you are currently working in.
You cannot copy data to all versions of a multi-version quoting policy transaction in one copy data action.
- **Copy data from** – You can copy data from any single version of a multi-version quoting policy transaction.

Copy Data and Side-by-Side Quoting Policy Transactions

In one copy data action, you can copy data to a single version side-by-side quoting policy transaction. However, if you use copy data on a side-by-side quoting policy transaction, any base data you copy to the version you are working with gets copied to the other versions. This behavior is the same as making a change to the base data in the PolicyCenter user interface: PolicyCenter copies that change to the other versions.

Including Child Entities When Copying Data

In some cases, you can include the children of an entity when copying data. For example, when copying a vehicle, you can choose to copy all or selected coverages. The vehicle details, except for assigned drivers, are copied by default.

Copy data always includes certain child entities when copying the entity. For example, copy data always includes the following:

- **Personal vehicle modifiers** – specify features of the vehicle such as whether the vehicle has a passive restraint system or anti-lock brakes. Copy data always copies vehicle modifiers because they are part of the vehicle.
- **Coverage terms** – Are always copied when copying a coverage.

Overview of Split and Spin-off Policies

PolicyCenter allows you to *split* an existing policy into two policies. PolicyCenter also allows you to *spin-off* a single policy from an existing policy.

In the base configuration, this functionality is available in the personal auto line of business. By configuring PolicyCenter, you can add this functionality to other lines of business.

Note: The ability to split and spin-off policies from an existing policy requires that copy data is configured for that line of business.

In personal auto, you may want to split or spin-off a policy for one of the following reasons:

- **Split** – A couple gets divorced. Both spouses wish to remain with the carrier. The carrier creates two new accounts, and splits the coverables on the existing policy into coverables on policies in the new accounts. The split creates two submission policy transactions. The carrier cancels the original policy.
- **Spin-off** – A son moves out of the house, and takes a car covered on his parents' policy. The carrier creates a new account for the son, and moves the car from the parents' policy to a new policy on the son's account. Spin-off creates a single submission. The carrier does not cancel the original policy after spinning-off part of the policy.

Splitting or spinning-off a policy has the following features:

- The data available to include on the split or spun-off submissions comes from the last slice of the bound policy.
- PolicyCenter creates a link between the source policy and any submissions or policies split or spun-off.
- The account that contains the split or spun-off policies can be the current account, a related account, or an arbitrary account.
- The producer of record and the producer of service on the submission are both set to the current producer of service on the policy. You can change both of these during completion of the submission.
- You can select the primary named insured from all named insureds on the account.
- The new submissions are of the same product as the source policy.
- You cannot create new submissions with a company contact as the primary named insured if the product does not support company contacts.

See also

- “Working with Split and Spin-off Policies” on page 309

Policy File Screens, Menus, and Actions

The policy file is the electronic file in which PolicyCenter stores policy information that is part of the legal contract. Depending on the task, PolicyCenter organizes policy information into different areas of the user interface. Understanding how PolicyCenter organizes this information allows you to quickly find policy information.

The following illustration shows some of the main areas of the policy file:

1. Info bar
2. Policy contract
3. Tools menu
4. Actions that may affect the current policy

Info Bar

Use the **Info Bar** to quickly view main information about the policy.

The first item indicates where you are. In this example, you are in the policy file. The second item displays the policy type. The third item displays the primary named insured for the policy. The fourth item displays the account number. If you click that link, PolicyCenter takes you to the **Account File** for the insured. The next item is

the policy number. Depending on where you are in the policy file, this too can be a link. The final item displays the policy status. In this example, you can see that the policy is in force and when it is due to expire. Other status messages include information on whether a submission needs approval, or who the underwriter is.

Actions Menu

Use the **Actions** menu to start an action to the policy. Actions on the policy include starting contextually appropriate policy transactions on the policy. The **Actions** menu is contextual and displays only the actions that apply to the policy at that time. For example, if a policy is scheduled for cancellation, then one of the options available to you would be to rescind the scheduled cancellation.

Policy Contract

This part of the policy file contains policy information that is part of the legally bound policy contract. The links in this section vary by line-of-business although all lines-of-business share certain steps, such as **Policy Info**, **Forms**, or **Payment**.

Policy Tools Menu

The **Tools** section contains links to supporting information that apply to the policy. This menu is context-sensitive.

Items in the **Tools** menu can include:

- **Summary** – View the contents of the policy file summarized on one screen. The screen also includes information regarding current activities, policy transactions, and policy transactions in progress.

The bottom of the **Summary** screen displays:

- **Current Activities**
- **Policy Transactions** – In the base configuration, PolicyCenter lists all the policy transactions executed against a policy by effective date in reverse chronological order. The **Type** column shows which policy transactions have been applied to a policy. The **Transaction #** column lists the policy transaction number associated with each policy transaction. Select either link to view the details.
- **Pending Policy Transactions**
- **Split Off Policies** – Contains links to policies or policy transactions that are the result of a split, spun-off, or rewrite to a new account.
- **Billing** – View overall balance and balances for individual policy periods. You can also view the payment schedule.
- **Contacts** – View contacts that have a role on the policy. For each contact, you can view basic contact information and the roles the contact plays on the policy.
- **Participants** – This screen lists the users that interact with the policy by the role that they perform on the policy. The screen also shows the assigned group that the user belongs to. You can add or remove participants. For existing roles, you can change the user who performs that role.
- **Notes** – Search and view notes related to the policy transaction or policy.
- **Documents** – Search and view any attached documents related to the policy transaction or policy.
- **Policy Transactions** – View summary information about all policy transactions that have occurred on the policy. The list includes policy transactions that have modified the policy and policy transactions that are in-progress, withdrawn, not taken, or non-renewed. You can also compare policy transactions.
- **Risk Analysis** – View any issues that may affect the policy. This section includes underwriting referral reasons, underwriting issues, claims, prior policies, or prior losses. You can also add, close, or reopen underwriting referral reasons.
- **History** – Search and view historical events that pertain to the policy. You can filter history events by the following:
 - **User**

- **Timestamp** – Specify **From** and **Until** dates.
- **Related To** – This drop-down list allows you to filter events for a specific policy transaction or to show all events.

The **History** screen displays the following information for history events: **Type**, **User**, **Event Timestamp**, **Description**, **Job**, **Original Value**, and **New Value**. In the base configuration, submission, renewal and other policy transactions log history events. For more information, see “Configuring Job History Events” on page 552 in the *Configuration Guide*.

Working with Policies

This topic describes how to work with policies in the PolicyCenter user interface.

Working with Copy Data

This topic explains how to copy data from one policy to another in the PolicyCenter user interface.

To copy policy data, you must have the **Copy policy data** permission. The code for this permission is `copypolicydata`.

To copy data from one policy to another

1. Start or navigate to a policy transaction and line of business that supports copy data. For example, in a personal auto submission policy transaction you can copy data from another policy.
2. Select **Actions** → **Copy Data**.

PolicyCenter displays the **Copy Policy Search Policies** screen. This screen allows you to search for policies or policy transactions to copy data from. By default, the **Account Number** field is set to the account number of the target policy.

3. Make a selection from the **Search For** drop-down list.

Select **Search For** → **Policy** to search for a policy. Enter information about the primary insured. You can also search on the account and policy numbers. You can specify an **As of Date** to constrain the search results to policy terms in effect on that date.

Select **Search For** → **Policy Change**, **Renewal**, **Rewrite**, **Rewrite New Account**, or **Submission** to search for a policy transaction. Enter information about the primary insured. You can also search on the account, policy, and policy transaction numbers.

PolicyCenter retrieves both open and completed policy transactions. Completed policy transactions include expired and withdrawn policy transactions. You can optionally select whether to search by effective date or creation date. You must specify whether you want to retrieve all works orders, or policy transactions within a date range. PolicyCenter returns a list of matching policy transactions in the specified date range.

Note: You can copy data from other policy periods or other policy transactions on the current policy.

4. Click **Search** to display the list of **Search Results**.

5. Click **Select** on one of the **Search Results**.

PolicyCenter displays the **Select data to copy from** screen. This screen allows you to choose which data to copy from the selected policy or policy transaction.

For policy transactions, PolicyCenter displays the slice on the edit effective date. For policy terms, PolicyCenter displays the last slice of the policy period. For policy terms, you can specify an **As of Date** on this screen. If you enter an **As of Date**, the screen displays the entities available to copy as of that date, or slice.

In the **Personal Auto Line** tab, you can select the following types of data to copy:

- **Drivers**

- **Vehicles** – The copy includes modifiers. You can choose to copy all or selected vehicle coverages. You can choose to copy additional interests.
 - **PA Coverages**
 - **Exclusions**
 - **Conditions**
6. Click the **Notes** tab to copy all or selected notes.

Working with Split and Spin-off Policies

You must have the **Divide Policies** permission to view **Split Policy into Two** and **Spin-off Policy from this One** from the **Actions** menu. The code for this permission is `dividepolicy`.

This topic provides step-by-step instructions for splitting a policy. The steps for spinning-off a policy are similar but create a single submission instead of two.

To split an existing policy into two policies

1. Navigate to an existing policy in a line of business that supports split or spin-off policies. In the base configuration, the personal auto line of business support split and spin-off policies.

2. Select **Actions** → **Split Policy into Two**.

PolicyCenter displays the **Split Policy** screen with **Submission #1** on the left and **Submission #2** on the right. Each submission has the following fields:

Field	Description
Account Number	Required. Click the account picker icon to choose an account by using the Search Accounts screen.
Name	After you select an Account Number , this field displays the name of the account holder on the selected account.
Quote Type	Required. Select Quick Quote or Full Application . Default value is Full Application .
Default Effective Date	Required. Default value is the current date.
Primary Named Insured	Required. Select the primary named insured for the policy. The selection lists all named insureds on the account. Default value is the account holder.
Select data to include on new submission	This section displays the policy data configured for copy data in the current line of business. (Notes are not available to copy when splitting or spinning-off policies.) In personal auto, you can select to include drivers, vehicles, coverages, exclusions, and conditions in the new submission.

3. Make your selections in each submission, and click **Create Submissions**.

PolicyCenter displays the **Split Policies Complete** screen. This screen has links to original policy and to the two submissions split off from it.

Configuring Policies

This topic describes how to configure policies.

Core Entities Associated with Policies

The entities at the core of the PolicyCenter data model are: **Policy**, **PolicyPeriod**, **Job**, **PolicyTerm**, and **PolicyLine**. The following sections provide descriptions of these and other entities.

- **Policy Object Model Overview**

- Policy Term and Policy Period Entity Relationship Diagram

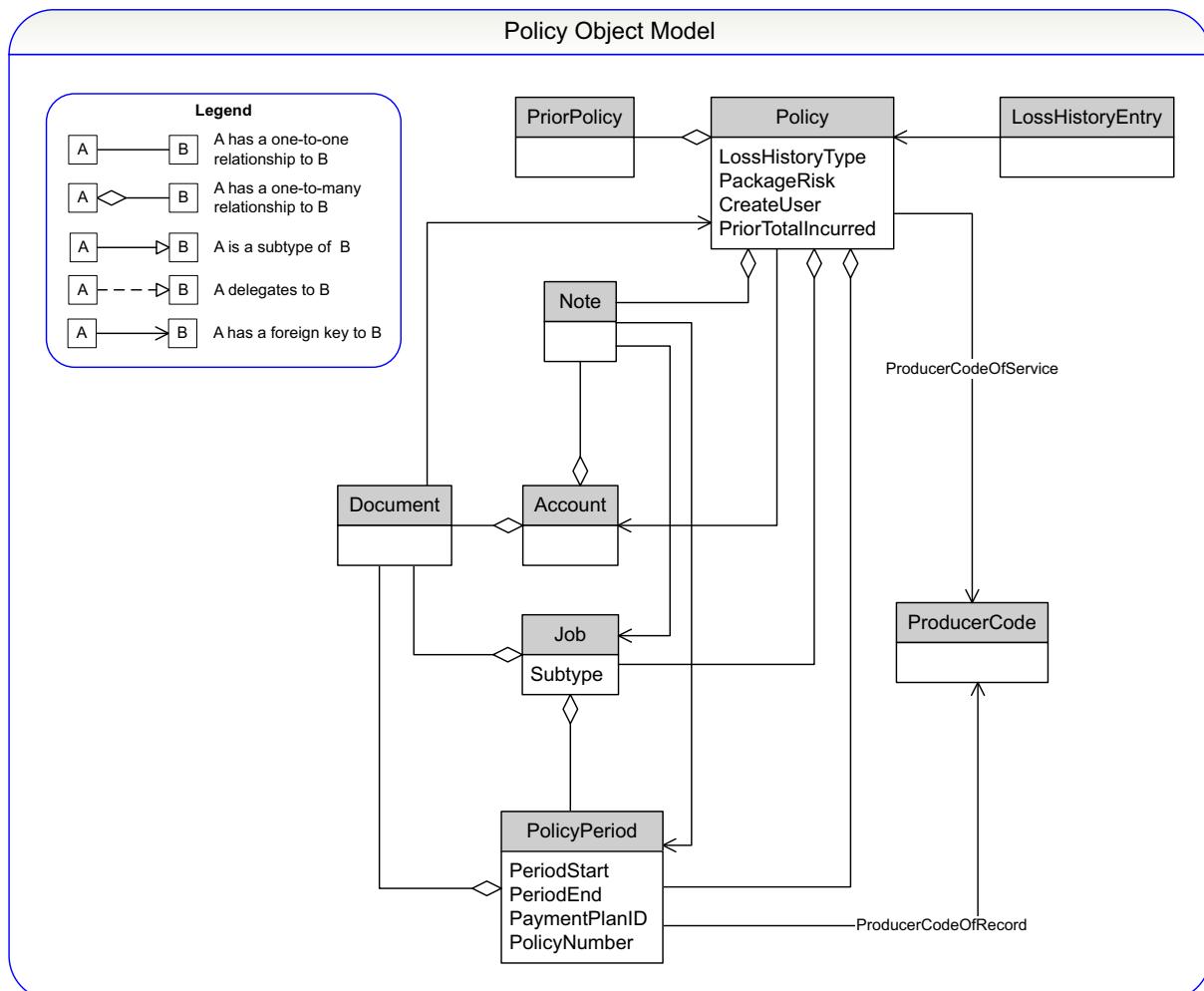
See also

- “Businessowners Object Model” on page 200
- “Commercial Property Object Model” on page 225
- “Commercial Package Object Model” on page 212
- “General Liability Object Model” on page 242
- “Inland Marine Object Model” on page 254
- “Personal Auto Object Model” on page 269
- “Workers’ Compensation Object Models” on page 295

Policy Object Model Overview

The following object model diagram shows some of the basic relationships of policy objects. This diagram focuses on the entities that interact with the `Policy` entity.

Note: See the *Data Dictionary* to see a complete list of all properties in the entities. The diagram and entity descriptions contain a partial list, highlighting those that may be of interest to you.



Policy Entity

A policy is a contract of insurance that describes the term, coverage, premiums, and deductible. A policy protects the insured from accidental loss. A policy also lists the people or properties being insured against loss. If a carrier offers a policy and an insured accepts the terms in the policy, it becomes bound and is an enforceable legal document. Policies are defined by dates or periods of time. For example, your auto policy is in force from January 1st to June 30th. These are called *policy periods*.

The Policy has access to individual note types through derived properties such as `creditworthyNotes` and `generalNotes`.

The main foreign keys to Policy are:

- `Document`
- `DocumentSearchCriteria`
- `Job`
- `LossHistoryEntry`
- `Note`
- `NoteSearchCriteria`
- `PolicyPeriod`
- `PriorPolicy`
- `UserRoleAssignment`

Job Entity

The Job entity contains these subtypes: `Audit`, `Cancellation`, `Issuance`, `PolicyChange`, `Reinstatement`, `Renewal`, `Rewrite`, and `Submission`. Each policy transaction processes a policy in a different way. The `Submission`, `Rewrite`, and `Renewal` jobs (policy transactions) create new policy periods and new policy terms. You can access all the jobs for a policy from the `Jobs` array.

Other Entities Associated with Policies

The Note and Document entities have foreign keys to Account, Policy, Job, and PolicyPeriod. The Message entity has foreign keys to Account, Policy, PolicyPeriod.

Entities such as Note, Message, and Document have foreign key references to Policy because they span the life of the policy, not just a time period.

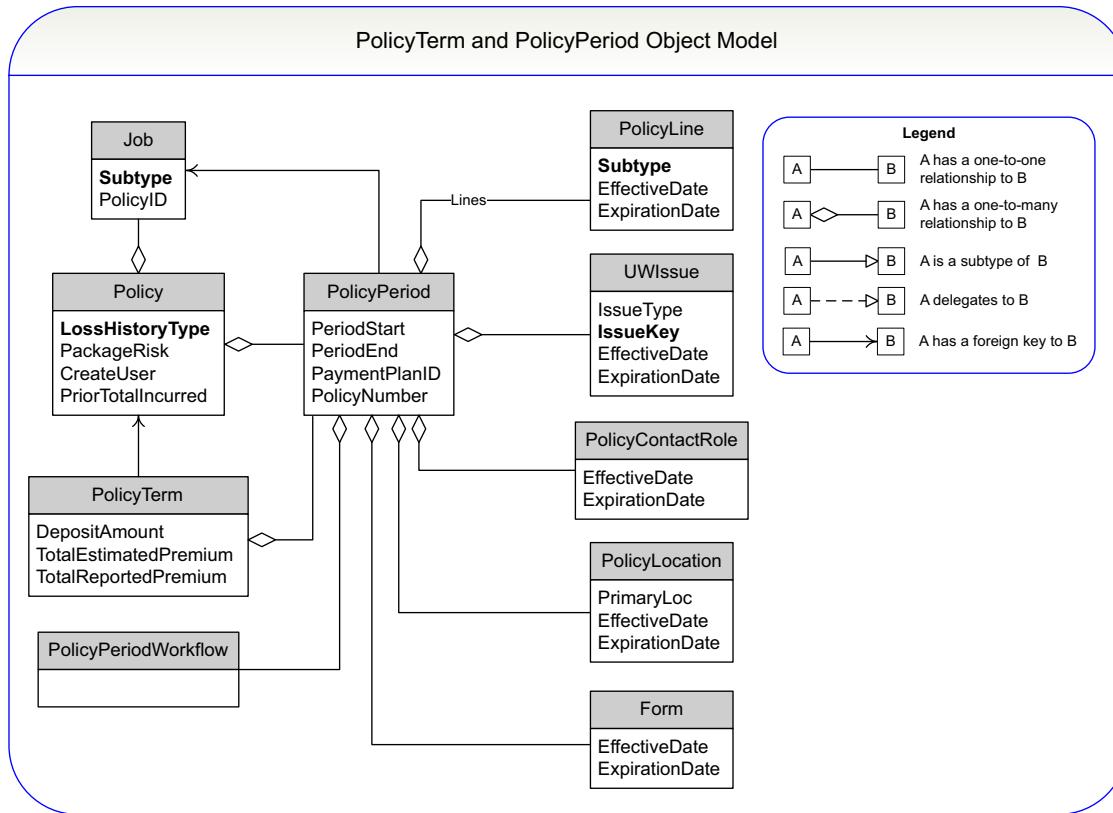
The PriorPolicy entity contains information about the prior policy.

The LossHistoryEntry entity contains information about prior policy losses.

The ProducerCode entity identifies the producer of service and has fields such as `ProducerStatus` and `Description`. It also has a foreign key to PreferredUnderwriter.

Policy Term and Policy Period Entity Relationship Diagram

The following illustration shows some of the key entity relationships for the **PolicyTerm** and **PolicyPeriod** entities. The **PolicyTerm** and **PolicyPeriod** entities represent different information about the contractual period of a policy.



Policy Term Entity

The **PolicyTerm** entity represents one contractual period for the policy. A policy term represents one contractual period for a policy. The contractual period extends from the date the policy goes into effect (the effective date) to the date it expires (the expiration date). For example, if a homeowners policy has a year long period starting on January 1, calendar year 2013 is one policy term.

There is only one **PolicyTerm** entity for each contractual period. The fields on the **PolicyTerm** entity apply to the whole contractual period. Unlike the **PolicyPeriod** entity, the **PolicyTerm** entity and its subentities are not revised.

PolicyCenter creates a new **PolicyTerm** whenever PolicyCenter completely recreates the policy contract. PolicyCenter completely recreates the policy contract with a new policy submission, a renewal, or a rewrite of an existing contract. PolicyCenter does not create a new **PolicyTerm** when you amend a policy contract with a policy change job. The **PolicyTerm** has a foreign key to the **Policy**. The **PolicyTerm** and **Policy** entities have arrays of **PolicyPeriod** entities.

Policy Period Entity

The **PolicyPeriod** entity stores information for a specific revision of the contractual period of a policy. A revision occurs anytime a job occurs on a policy. A submission creates the first revision. Each additional transaction on the policy (such as a policy change) creates a new revision. Therefore, a policy almost always has multiple revisions, with one **PolicyPeriod** entity for each revision. During the contractual period, only one

PolicyPeriod entity is in effect at a time. The PeriodStart and PeriodEnd properties contain the start and end dates of the contractual period.

Each PolicyPeriod entity is the root of a complex graph of subentities such as policy lines, vehicles, coverages, and many others. These subentities have EffectiveDate and ExpirationDate fields which specify when the entity becomes effective and when it expires. The EffectiveDate and ExpirationDate are bounded by the contractual period (represented by the PeriodStart and PeriodEnd fields on the PolicyPeriod entity).

PolicyCenter creates a new revision when you process a policy change that adds a car to the policy. The EffectiveDate for the car is several months into contractual period, and the ExpirationDate extends to the end of the period. PolicyCenter clones a new PolicyPeriod entity and its subentities and adds an entity for the new car. The contractual period now has two PolicyPeriod entities. The new PolicyPeriod entity is in effect. PolicyCenter preserves the old PolicyPeriod entity as a historical record of the policy.

Each PolicyPeriod has a Status which is a typekey to PolicyPeriodStatus. The PolicyPeriodStatus typecodes include values such as Binding, Canceling, Quoted, and Withdrawn. See the *Data Dictionary* for the complete list of typecodes.

When you start a new policy transaction (job) or create a new revision, PolicyCenter creates a PolicyPeriod. For a short amount of time during initialization, the policy period is in Temporary status. However, if an error occurs during initialization of the job or policy period, the policy period can remain in Temporary status. This policy period may persist in the database. Because the initialization did not complete, a policy period that is in Temporary status may contain invalid data. In your code, be sure to check for the Temporary status on PolicyPeriod and to avoid using data from policy periods with this status.

See also

- “Policy Revisioning” on page 479

Policy Line Entity

A policy can be monoline or multi-line. A monoline policy contains a single type of insurance, such as personal auto. A multi-line policy contains more than one type of insurance, such as a commercial package policy with property and general liability. PolicyLine contains subtypes which include: BusinessOwnersLine, PersonalAutoLine, and WorkersCompLine.

The PolicyPeriod entity has boolean fields (such as BOPLineExists or CPLineExists) for each policy line. The boolean field indicates whether or not that policy line exists on the policy period. If the policy line exists, then the BOPLine or CPLine field, for example, allows you to access the policy line.

Workflow Entity

The Workflow entity has more than one subtype, but the one that pertains to PolicyPeriod is PolicyPeriodWorkflow. PolicyPeriodWorkflow has a foreign key to the policy period associated with this workflow.

Job Entity

The Job entity has subtypes of Audit, Cancellation, Issuance, PolicyChange, Reinstatement, Renewal, Rewrite, and Submission. It contains foreign key references to Policy and other entities.

Configuring Copy Data for a Line of Business

In the base configuration, copy data is available in the personal auto line of business. You can modify the personal auto line to meet your business needs. You can also configure copy data in other lines of business.

Copy data enables you, the user, to quickly and accurately take information from one policy and copy it to a policy transaction for another policy. In addition, you can copy data from prior versions or policy transactions of a policy to the current policy transaction. Copy data enables copying information from a source period into a target period by providing the following:

- A mechanism for searching and selecting the source period.
- A user interface that controls which items to copy from the source period.
- Copier and CompositeCopier classes that copy the information from the source period data into the target period. These classes are in the `gw.api.copy` namespace.

See also

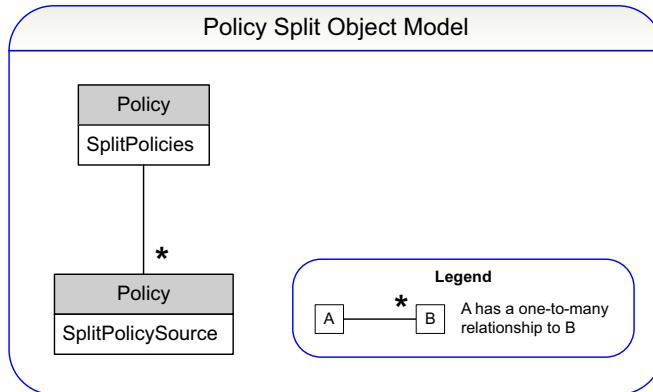
- “Configuring Copy Data in a Line of Business” on page 191 in the *Product Model Guide*

Configuring Split and Spin-off Policies

This topic describes how to configure split and spin-off policies. A line of business that provides the ability to split and spin-off policies requires that copy data be configured for that line of business.

Split and Spin-off Policies Object Model

A source policy and the policies split or spun-off from it are connected by fields in the object model. The `SplitPolicies` array on the source policy provides access to policies split or spun-off from the source policy. The `SplitPolicySource` foreign key points from a split or spun-off policy back to the source policy. A split or spun-off policy can have only a single source policy.



Split and Spin-off Methods in Gosu Classes

The `gw.product.DividePolicySelection` Gosu class collects basic information for creating the submission for a split policy. The basic information includes a `ProducerSelection` object, the `QuoteType`, and an `AccountContact` to create the `PrimaryNamedInsured`.

After collecting the basic policy information, the `createSubmission` method creates a new submission. Next in the `initializeSubmission` method, a `PolicyPeriodCopier` object copies policy data to the submission. Split policies have two `DividePoliciesSelection` objects, one for each submission. These objects are independent and are not directly connected.

Archiving in PolicyCenter

Archiving is the process of moving the data associated with a policy from the active PolicyCenter database to a document storage area. In PolicyCenter, you archive a policy term. In turn, PolicyCenter archives the data associated with all policy periods in that policy term. You can search for archived policy terms then request that PolicyCenter retrieve and restore an archived policy term from the archive. While archived, the data associated with the policy term occupies less space in the active database.

This topic includes:

- “Archiving Overview” on page 315
- “Archiving Policy Terms” on page 316
- “Searching for Archived Policy Periods” on page 319
- “Retrieving Archived Policies” on page 320
- “Configuring Policy Archiving” on page 321
- “More Information on Archiving” on page 321

Archiving Overview

Policy term archiving in PolicyCenter has the following characteristics:

- PolicyCenter only archives policy terms that have been expired for a specified number of days. You specify the number of days in a configuration parameter.
- A batch process automatically identifies policy terms that need archiving, and then archives them.
- The archiving process may skip or exclude some policy terms based on:
 - If the policy terms are still associated with other active objects, such as open policy transactions (jobs)
 - If the policy terms were recently retrieved from the archive
 - If a user requested the policy be excluded from archiving

In PolicyCenter, you can search for archived policies on the screen that you search for policies. For both active and archived policies, the **PolicyTerm** object is the basis of the search.

- In PolicyCenter, you can search for archived policies on the screen that you search for policies. For both active and archived policies, the `PolicyPeriod` object is the basis of the search.
- You can choose to search for archived policies and related entities. However, such searches depend on fields present in the active database. Therefore, some search fields, such as those on most effective-dated entities, are inappropriate when searching and including archived terms.
The policy search results display the same policy summary information whether the policy is active or archived. Archived policies appear but are not selectable.
You can configure policy search to use `Policy`, `PolicyTerm`, `PolicyPeriod`, or even deeper objects as the basis of the search.
- A user can request that PolicyCenter retrieve an archived policy at any time. A batch process retrieves the policy, restores it to the PolicyCenter database, and sends an activity to the user.
- The retrieved policy is identical to a policy that has never undergone the archiving process.

See also

- “Configuring Policy Archiving” on page 423 in the *Configuration Guide*
- “The Domain Graph” on page 245 in the *Configuration Guide*
- “Archive Parameters” on page 36 in the *Configuration Guide*
- “Domain Graph Parameters” on page 49 in the *Configuration Guide*
- “Info Pages” on page 128 in the *System Administration Guide*

Advantages of Archiving

The main advantage of archiving policy periods is to improve the operational performance of the application. As PolicyCenter creates more policy periods, database storage requirements increase, table lengths increase, and performance degrades. Archiving can improve performance in the face of unbounded policy period growth.

If you have a legacy system that contains many policy terms, then you can add them to your main database, attach them to accounts, and immediately archive them. In PolicyCenter, you can search for these legacy policy terms and retrieve them from the archive.

Impact of Archiving on Your PolicyCenter Configuration

Enabling archiving may have a significant impact on your PolicyCenter configuration and integration. Archived policy data is no longer immediately available in the active database. Your implementation of archiving may require data model changes.

If you are configuring a PolicyCenter implementation that may implement archiving in the future, Guidewire suggests that you:

- Evaluate your data model against the requirements for archiving.
- Understand the implications of archived data on PCF files, code, and database searches.

Archiving Policy Terms

In PolicyCenter archiving, a complete `PolicyTerm` (including all of its `PolicyPeriods`) is the logical unit of archiving.

PolicyCenter archives a policy term if the term is canceled or no longer in effect and has been inactive for a specified period of time. When PolicyCenter archives the policy term, it marks the policy term as archived, and archives each policy period in the term.

Because the logical unit of archiving is the term, it is possible that one or more periods were not archived or retrieved. In PolicyCenter, a given policy period and its graph are considered archived if one or more of the periods in the term is archived.

When PolicyCenter archives a policy period, it sets an archive flag on the policy period and archives related information. The related information is information in the domain graph and other related entities such as underwriting issues, notes, and documents associated with the policy period. The archived entities do both of the following:

- Have a path to the `PolicyPeriod` object by means of foreign key relationships.
- Delegate to the `Extractable` entity

In some cases such as `Note`, the entity delegates to the `Extractable` entity directly.

In an `EffDated` entity such as `PolicyDriver`, the entity delegates to `Extractable` through inheritance. The `EffDatedBase` entity delegates to `Extractable`.

See also

- “The Domain Graph” on page 245 in the *Configuration Guide* for an explanation of what the domain graph is and how to use it.

Policy Terms that PolicyCenter Does Not Archive

If a policy term meets the conditions for archiving, all policy periods in that term must meet additional criteria. The following table contains a list of conditions that can prevent PolicyCenter from archiving a policy period. PolicyCenter does not archive a policy term if one or more policy periods in that term prevent it from being archived.

Policy periods that cannot be archived	Reason
Policy periods with open or recently completed policy transactions	The policy period must not have a policy transaction that completed within <code>ArchiveRecentJobCompletionDays</code> .
Policy periods that are part of an unfinished workflow	It is not possible to close a policy period that has an active workflow. PolicyCenter only attempts to archive a closed policy period.
Policy periods with audits and premium reports scheduled, in progress, or recently completed	The policy period must not have audits and premium reports completed within <code>ArchiveRecentJobCompletionDays</code> .
Policy periods that have as a <code>basedOn</code> ancestor policy period with an open policy transaction.	Because an open policy transaction might cause changes that need to be migrated forward, PolicyCenter does not archive future terms. Conversely, you cannot start a policy transaction on a policy period that has later terms archived.
Policy periods with open claims	In the default configuration, PolicyCenter uses the <code>IPCArchivingPlugin</code> to determine if there are open claims.
Policy periods with open activities	The archive work item writer does not pick up policy periods with open activities for the following reason. If the writer archives a policy period with open activities, those activities disappear from the user's <code>Desktop</code> . The user cannot find or close those activities until PolicyCenter retrieves and restores the policy period from the archive. An open activity results in a delay of <code>ArchiveDefaultRecheckDays</code> .
Policy periods with pending messages	The policy period cannot contain messages that have been sent. It is unlikely that an old, canceled policy period is in this condition. If it is, the archiving batch process skips this policy period, and retries later until it finds that there are no more active messages. Pending messages result in a delay of <code>ArchiveDefaultRecheckDays</code> .
Policy periods excluded from archiving	The writer does not process policy periods marked as excluded. In the default configuration, you can exclude a policy from archiving by suspending it from archiving. PolicyCenter sets the <code>ExcludedFromArchive</code> bit set to <code>true</code> for a suspended policy period.

Entities Retained After Archiving

When PolicyCenter archives a policy term, PolicyCenter does not archive some entity instances that are part of the policy term. PolicyCenter archives but also retains some other entity instances.

PolicyCenter does not archive the `PolicyTerm` object.

The `PolicyPeriod` and associated instances of the `EffectiveDatedFields` entity are written to the archive, but also retained in the active database. These instances are available for read-only tasks, such as display and searches.

PolicyCenter archives but also retains `Note`, `Document`, and `UWIssueHistory` entity instances. After archiving, PolicyCenter removes the following retained entity instances from the PolicyCenter database:

- PolicyCenter removes `Note` entity instances associated with the policy transaction of the archived policy period if the `Note` does not have an associated `Activity`.
- PolicyCenter removes `UWIssueHistory` entity instances for auto-approvable issues.

Running the Archive Policy Terms Batch Process

This topic provides step-by-step instructions on how to archive policy terms in PolicyCenter. PolicyCenter archives a canceled policy term that has been inactive for a specified period of time.

This example assumes that you have canceled the Ray Newton personal auto policy. For instructions on how to cancel a policy, see “Canceling a Policy” on page 112.

This example also assumes the following:

- You have turned on archiving by setting the `ArchiveEnabled` parameter to true in `config.xml`.
- You have not changed the settings of any other archive parameters in `config.xml`.
- You have loaded the small sample data set. For more information, see “PC Sample Data” on page 139 in the *System Administration Guide*.

To archive policy terms

1. Press **Alt+Shift+T** to access the internal tools.
2. Advance the system clock by two years.
 - a. Select **Internal Tools** → **Testing System Clock**.
 - b. Click **Add Year** twice.
- In the default configuration, 366 is the minimum number of days that must pass before PolicyCenter can archive a policy period associated with a policy term. To trigger archiving, you must advance the system clock past this number of days.
3. Return to the **Batch Process Info** screen by selecting **Server Tools** → **Batch Process Info**.
4. Before archiving policy terms, you must run several batch processes to advance and remove workflows. In the default configuration, these include:
 - Purge Workflow – Purge completed workflows.
 - Purge Workflow Logs – Purge completed workflow logs.
 - Workflow – Advance any active workflows waiting on expired time-outs.

- Premium Ceding – Cede reinsurance premiums for reinstatement policy transactions. Because Premium Ceding points to PolicyPeriod, any rows in its WorkItem table must be processed.

Note: It is only necessary to run these batch processes while you are testing archiving and artificially moving the system clock. In a normal production environment, these batch processes run regularly according to schedule. In the default configuration, archiving considers policy transactions that have been canceled for at least a year. During the year, these batch processes will run at least once.

5. Run the Archive policy terms batch process.
6. Select Actions → Return to PolicyCenter.
7. To view what remains in PolicyCenter of the archived Ray Newton policy, see “Searching for Archived Policy Periods” on page 319.

Searching for Archived Policy Periods

In PolicyCenter, you can search for archived policy periods alongside active ones. The archived policy period appear but are not selectable in the search results. If you click to view a policy period, PolicyCenter displays the policy on the summary screen with the label **Summary (Archived)**. Under **Policy Transactions**, the archived policy terms are not selectable.

Example of Searching for Archived Policy Periods

This topic provides step-by-step instructions on how to search for archived policy periods in PolicyCenter.

This example assumes that you have:

1. Canceled the Ray Newton personal auto policy. For instructions on how to cancel a policy, see “Canceling a Policy” on page 112.
2. Advanced the system clock so that the Ray Newton policy is eligible for archiving and run the archiving batch process. For instructions, see “Running the Archive Policy Terms Batch Process” on page 318.

This example also assumes the following:

- You have turned on archiving by setting the `ArchiveEnabled` parameter to true in `config.xml`.
- You have not changed the settings of any other archive parameters in `config.xml`.
- You have loaded the small sample data set. For more information, see “PC Sample Data” on page 139 in the *System Administration Guide*.

To search for archived policy terms

1. In PolicyCenter, click the **Search** tab.
2. On the **Search Policies** screen, select **Include Archived** in the **Source** field.
3. In **Producer Code**, enter 100-002541, then click **Search**.

In **Search Results**, notice that a number of policies are archived. The archived policies appear in the search results but are not selectable.

4. Select the Ray Newton personal auto policy.

PolicyCenter displays the policy on the summary screen with the label **Summary (Archived)**. Under **Policy Transactions**, the archived policy terms appear but are not selectable.

Desktop and Team Tabs and Archiving

In the default configuration, the filters on the **Desktop** and **Team** tabs do not return items stored in the archive. This behavior is because the **Desktop** and **Team** screens shows current items that are likely to need the user's attention.

You can modify the screens in the **Desktop** or **Team** tabs to display archived items. If you choose to display archived items, modify the desktop or team PCF files to check the archive status of the items.

See also

- “Gosu Code and Archiving” on page 430 in the *Configuration Guide*

Retrieving Archived Policies

If a policy is archived, you can only see a summary of the policy. To see the whole policy again, you can retrieve it from the archive and restore it to the PolicyCenter database. PolicyCenter restores the policy in its entirety.

When PolicyCenter retrieves a policy, PolicyCenter generates an activity for the user or users who requested retrieval of that policy.

To retrieve an archived policy, you must have the **Retrieve from archive** permission on the policy. You must also have both view and edit permissions on the policy.

Example of Retrieving an Archived Policy

This topic provides step-by-step instructions on how to retrieve a policy from the archive and restore it to PolicyCenter.

This example assumes that you have canceled the Ray Newton personal auto policy. For instructions on how to cancel a policy, see “Canceling a Policy” on page 112.

This example also assumes the following:

- You have turned on archiving by setting the `ArchiveEnabled` parameter to true in `config.xml`.
- You have not changed the settings of any other archive parameters in `config.xml`.
- You have loaded the small sample data set. For more information, see “PC Sample Data” on page 139 in the *System Administration Guide*.

To request retrieval of a policy

1. Navigate to an archived policy, such as the Ray Newton policy you archived in “Running the Archive Policy Terms Batch Process” on page 318.

PolicyCenter displays the **Summary (Archived)** screen for that policy.

2. Click **Request Retrieve**.

PolicyCenter displays the **Request Retrieve from Archive** screen.

3. Optionally, enter a **Reason** then click **Request Retrieve**.

To retrieve archived policies

1. Login as su.
2. Press **Alt+Shift+T** to access the internal tools.
PolicyCenter displays the **Batch Process Info** screen.
3. Run the **Retrieve policy terms** batch process.
4. Select **Actions → Return to PolicyCenter** and view the retrieved policy.

PolicyCenter generates an activity notifying the user that the policy has been retrieved. PolicyCenter generates an activity each time the user presses the Request Retrieve button.

Configuring Policy Archiving

For information on how to configure archiving for your business practices, see the following:

- See “Configuring Policy Archiving” on page 423 in the *Configuration Guide* for a discussion of the different configuration points associated with archiving.
- See “The Domain Graph” on page 245 in the *Configuration Guide* for information on how PolicyCenter builds the domain graph.

More Information on Archiving

See also

- “Archiving in PolicyCenter” on page 315 in the *Application Guide* – information on archiving policies, searching for archived policies, and retrieving archived policies.
- “Archive Parameters” on page 36 in the *Configuration Guide* – information on the configuration parameters used in archiving.
- “Configuring Policy Archiving” on page 423 in the *Configuration Guide* – information on configuring archiving, selecting entities for archiving, and archiving and the object (domain) graph.
- “Archiving Integration” on page 613 in the *Integration Guide* – describes the archiving integration flow, storage and retrieval integration, and the plugin interfaces.
- “Logging Successfully Archived Policy Terms and Policy Periods” on page 23 in the *System Administration Guide*.
- “Archive Info” on page 129 in the *System Administration Guide*.
- “Upgrading Archived Entities” on page 172 in the *Upgrade Guide*.

Account File

In PolicyCenter, you can view and manage account information separately from policy transactions such as submissions, renewals, or policy changes. PolicyCenter provides a complete view of the account, where you can view and edit account information. In the account file, you can access information about the primary insured, related contacts, location data, policies, policy transactions, and producer codes.

This topic includes:

- “Account Overview” on page 323
- “Account Screens” on page 325
- “Account Actions” on page 329
- “Working with Accounts” on page 329
- “Configuring Accounts” on page 335
- “Configuring Moving Policies Between Accounts” on page 337
- “Configuring Account Relationships” on page 337

Account Overview

In PolicyCenter, you have the flexibility to create and manage account information as a separate process from managing a policy. PolicyCenter is flexible in account creation. You can create an account from the **Account** tab or the **Desktop Actions** menu.

This topic describes some of the features of accounts.

Account Security Overview

In PolicyCenter, you can restrict who sees an account. Typically, PolicyCenter limits account visibility to users having one of the producer codes associated with the policies on the account.

- If a user has a producer code that is the producer of service for a policy on the account, then that user usually has access to the account. This access includes view and edit permissions that the producer code provides.

- Producers of record usually do not have any account-level access. PolicyCenter limits them to see only information on their own policies. This feature is configurable.

Users, such as producers of record, who have access to a policy, but not to the account, typically do not have access to the links to the account file. Based on the security configuration, users may still have visibility to some account-level information through the policy.

See also

- “Security Restrictions Using the Status Field” on page 643 for information on producer codes security.

Related Accounts Overview

In PolicyCenter, you can associate accounts with one another. In the base configuration, the account relationships are:

- Parent and child** – Use this account relationship for hierarchical accounts, such as a corporate parent and subsidiaries.
- Common owner** – Use this account relationship to link commercial accounts that have a common owner. The common owner might be a person or a corporate entity. For example, you can associate all accounts for companies owned by one holding company, even though there is no account for the parent company.

You can modify the existing account relationships or create your own account relationship types.

With account relationships, you can also search for accounts with a shared contact. In the base configuration, this search finds accounts that have an account holder or named insured in common. The contact in common does not have to be a named insured or account holder on both accounts. For example, if a contact is a account holder on one account and a named insured in another, these are related accounts. You can modify or add to the search criteria.

See also

- “Searching for Accounts with a Shared Contact” on page 334
- “Configuring Shared Contact Search Criteria” on page 341

Overview of Moving and Rewriting Policies Between Accounts

In PolicyCenter, you can move or rewrite policies from a source account to a target account. The following table compares moving and rewriting policies.

Moving	Rewriting
Moves all policy terms, policy transactions, and everything else associated with the policy, including activities, notes, and documents. Copies account contacts and locations referenced by the policy to the new policy. PolicyCenter removes the policy from the source account.	Moves the policy going forward to a target account, but the previous policy terms stay with the source account.
The move does not affect the in-force status of the policy.	The rewritten policy is never in-force simultaneously on both the source and target accounts.
Is done immediately.	Creates a policy transaction (job) that a user must complete. The code for this job subtype is RewriteNewAccount.
You can also move an in-progress submission or rewrite new account policy transaction.	Only issued policies can be rewritten to a new account.

See also

- “Rewrite New Account” on page 75
- “Rewrite New Account Policy Transaction” on page 135

Overview of Merging Accounts

In PolicyCenter, you can merge an account (source) into another account (target). When you merge, PolicyCenter moves the policies, policy transactions, notes, activities, and other data from the source account to the target account. When you merge two accounts, only the target account remains. In the database, PolicyCenter marks the source account as frozen.

An underwriter may need to merge two accounts into one if two accounts represent the same person or company. This situation may occur because of an error such as a misspelled name, or when bringing in accounts from a legacy system.

When searching for an account to merge, you can search for:

- Accounts related to the target account
- Any account for which you have permissions

The account holder type must be the same on both the source and target accounts. You cannot merge a personal account into a company account.

Access to Producer Codes Required for Merging Accounts

On the source account, you must be assigned to the producer code of service on every policy.

On the target account, you need to have access to the account. This access requires that you be assigned to at least one of the producer codes listed on the account. Therefore, you must have access to the producer code of service for at least one policy on the target account.

See also

- “Merging Accounts” on page 333

Account Screens

This topic describes screens for accounts.

The account menu links in the left sidebar provide supporting information for the account. This menu is context-sensitive.

- “Account File Summary Screen” on page 326
- “Account File Contacts Screen” on page 326
- “Account File Locations Screen” on page 326
- “Account File Participants Screen” on page 327
- “Account File Policy Transactions Screen” on page 327
- “Submission Manager Screen for Accounts” on page 327
- “Underwriting Files Screen for Accounts” on page 327
- “Account File Related Account Screen” on page 327
- “Account File Documents Screen” on page 327
- “Account File Notes Screen” on page 327
- “Account File Claims Screen” on page 327
- “Billing Screen for Accounts” on page 328
- “Account File History Screen” on page 329

Account File Summary Screen

The **Account File Summary** screen for account displays the following information:

- Basic information such as account name, address, emails, phone numbers, and official IDs.
- **Status** shows the status of the account. The status can be:
 - **Pending** indicates that the account is ready for data entry, or has data but does not yet have any submissions. All new accounts begin with a status of pending.
 - **Active** means submissions have been created for it. The status changes to active when there is a bound policy on the account.
 - **Withdrawn** accounts are accounts that the carrier has withdrawn from consideration for business.
- **Current Activities** displays a list of current activities on the account and on policy transactions in the account.
- **Policy Terms** lists all policy terms associated with the account. Each policy term displays a policy number, policy type, status, effective date, and expiration date.
- **Pending Policy Transactions** initially displays open policy transactions. A menu item allows you to display all or complete policy transactions.

Multicurrency Fields

If multicurrency display is enabled, the **Account File Summary** screen has two multicurrency fields under the **Currencies** label. The **Settlement** field displays the account's preferred settlement currency (`Account.PreferredSettlementCurrency`). The **Coverage** field displays the account's preferred coverage currency (`Account.PreferredCoverageCurrency`). For either field, you can select one of the currencies configured in the base application.

Some account screens include summary account information which sums up monetary amounts from multiple policies. For example, the **Account → Billing** screen displays **Account Balances** fields such as **Billed Outstanding** and **Past Due** in the preferred settlement currency of the account. These fields provide an agent a snapshot that approximates the balances on the account, rather than the exact amount that the insured owes in a given currency. Therefore, in the base configuration, the amounts are converted without recording the exchange rates. The screen displays these sums in the preferred settlement currency of the account. If the policy amount is in another currency, PolicyCenter uses the current exchange rate to convert the amount to the preferred settlement currency on the account.

Account File Contacts Screen

The **Account File Contacts** screen lists contacts that you associate with the account and with the policies in the account. You can create contacts on the account and use these contacts in policies.

Additionally, contacts can have roles that have meaning only at the account level. A contact who has the role of driver on a personal auto policy, has an **MVR Report Details** button. Click this button to view all MVR reports received for that driver. For more information about motor vehicle reports, see “Motor Vehicle Records in Personal Auto” on page 258.

You can enter contact information on the account and access it when creating a submission or modifying a policy. See “PolicyCenter Contacts” on page 369 for details.

Account File Locations Screen

You can create locations on the account. Jobs can use these account locations. For example, you can access account location information when creating a submission or modifying a policy. See “Locations” on page 343 for details.

Account File Participants Screen

The **Account File Participants** screen lists the users that interact with the account by the role that they perform on the account. The screen also shows the assigned group that the user belongs to. You can add or remove participants. For existing roles, you can change the user who performs that role. Default types of roles include: **Requestor**, **Creator**, **Processor**, **Producer**, and **Underwriter**.

Account File Policy Transactions Screen

On the **Account File Policy Transactions** screen, you can view summary information about all policy transactions that have occurred on the account. From this screen, you can jump to the associated policy transaction or policy. You can filter the policy transactions by the following:

- Whether the policy transaction is open or complete.
- Job type, such as submission or renewal.
- Line of business.

Submission Manager Screen for Accounts

On the **Submission Manager** screen you can view submissions on the account. You can edit the submission if it is not complete. Additionally, incomplete submissions have an **Actions** menu that allows you to **Withdraw**, **Decline**, or mark the submission as **Not Taken**.

For each submission, you can create confirmation letters. There is a button that allows you to create new submissions.

Underwriting Files Screen for Accounts

On the **Underwriting Files** screen, you can view underwriting files on the account.

Account File Related Account Screen

On the **Account File Related Accounts** screen, you can view related accounts listed by **Relationship**, **Account Number**, **Name**, and **Address**. You can add and remove related accounts. You can also search for accounts with a common account holder or named insured.

Account File Documents Screen

On the **Account File Documents** screen, you can search for and view any attached documents related to the account or policies in the account. You can attach documents to the account. All documents attached to policies are also visible at the account level.

Account File Notes Screen

On the **Account File Notes** screen, you can search for and view notes related to the account or policies in the account. In PolicyCenter you can search for all notes on an account without knowing the policy on which the note was originally created.

Account File Claims Screen

On the **Account File Claims** screen, you can search for claims on the account. You can filter the search results by policy period or product.

Billing Screen for Accounts

In PolicyCenter, the **Billing** screen for an account displays account fields maintained by the billing system. For each account, you can view this page by clicking **Billing** in the left sidebar. This screen contains the following information:

- **Billing Account** – Displays the billing account. You can also display billing subaccounts of the current account. Use the drop-down list to select a subaccount and view the details of that account.
- **Account Balances** – Displays **Billed Outstanding**, **Unbilled**, and **Unapplied Funds**.
- **Collateral** – An additional asset or amount that may be required of an insured to secure coverage for a new or renewed policy. The insured can satisfy the collateral requirement with either cash, letters of credit (LOC), or a combination of both.
- **Primary Payer** – Displays the **Name**, **Address**, and **Phone** of the primary payer.

In the default configuration, PolicyCenter does not have a screen for specifying a billing subaccount. Billing subaccounts are retrieved from a billing system. In the default configuration, the `StandAloneBillingSystemPlugin` simulates retrieving billing subaccounts from a billing system. The large sample data set has examples of billing subaccounts. For information on how to load the sample data, see “[Installing Sample Data](#)” on page 52 in the *Installation Guide*.

IMPORTANT Billing subaccounts are different from the parent and child account relationships that you can define on the **Account File Related Accounts** screen.

The **View In BillingCenter** link allows you to view billing account details in BillingCenter. If you are logged into BillingCenter, the link jumps directly to the account. Otherwise, you go to a login screen. After logging in, BillingCenter displays the account.

In BillingCenter, you can view billing details. If you have sufficient permissions, you can start a delinquency or log a trouble ticket.

Policy Terms Tab

The **Policy Terms** tab at the bottom of the screen displays summary information for individual policy terms.

The **Owned Policies** section displays policies owned by this account. The summary information includes the policy number, effective dates, billing method, alternate billing account, balances, and invoice stream for individual policy terms.

The **Other Policies Billed to this Account** section displays policies billed to this account but not owned by this account. The summary information includes the policy number, effective dates, owning account, balances, and invoice stream.

Invoices Tab

The **Invoices** tab displays invoices retrieved from the billing system. You can choose to display invoices for the last three, six, and 12 months. For each invoice, the summary information includes statement and due dates, invoice number, invoicing period and payment instrument, status, and balances.

See also

- “[Billing System Integration](#)” on page 701

Account File History Screen

On the **Account File History** screen, you can view history events associated with the account and policies in the account. You must have the `accounthistory` permission to view this screen. The page initially displays all events related to the account. You can filter the history events by:

- User
- Timestamp – Specify **From** and **Until** dates.
- Related To – This drop-down list allows you to filter events for a specific policy transaction or policy, or to show all events for the account.

For more information about history events, see “Configuring Job History Events” on page 552 in the *Configuration Guide*.

Account Actions

The choices in the **Actions** menu for the Account File depend upon the current status of the account and user permissions. The following table describes each **Actions** menu choice. The marked cells indicate whether the menu choice is available for:

- My Accounts screen
- Accounts with active status
- Accounts with pending status
- Accounts with withdrawn status

Actions menu choice	My Accounts	Active	Pending	Withdrawn
New Submission – Navigates to the New Submission screen.	●	●	●	
New Account – Navigates to the Enter Account Information screen.	●			
New Note – Enter a new note at the bottom of the screen.		●	●	
New Document – Select Link to an existing document or Create a new document from a template.	●		●	
New Email – Enter a new email at the bottom of the screen.	●		●	
New Activity – From this choice, you can select Interview, New mail, Reminder, or Request.	●		●	
Withdraw Account – Sets the account status to Withdrawn.				●
Re-open Account – Sets the account status to Pending.				●
Move Policies to this Account – Select one or more policies to move to this account.	●		●	
Rewrite Policies to this Account – Select one or more policies to rewrite to this account.	●		●	
Merge Account into this Account – Merge another account into this account.	●		●	

Working with Accounts

This topic provides step-by-step instructions for accessing and working with accounts. Topics include:

- “Searching for an Account” on page 330
- “Creating an Account” on page 330
- “Tracking Your Accounts” on page 331

- “Moving a Policy From One Account to Another” on page 331
- “Rewriting Policies From One Account to Another” on page 332
- “Merging Accounts” on page 333
- “Adding an Account Relationship” on page 334
- “Modifying an Account Relationship” on page 334
- “Removing an Account Relationship” on page 334
- “Searching for Accounts with a Shared Contact” on page 334

Searching for an Account

To find an account, either:

- Navigate to the **Account** tab and click the account if it is visible or enter the account number in the **Acct#** field.
- Navigate to the **Search** tab and select **Accounts**. Enter your search criteria and click **Search**.

Minimum Search Criteria

When searching for an account, you must enter at least one of the following minimum search criteria:

Search Accounts field	Property on Account
Account Number	AccountNumber
Phone	WorkPhone
Tax ID	NameCriteria.OfficialID
Producer Code	ProducerCode
City and State/Province	City and State
Postal Code	PostalCode

Minimum search criteria is defined in the `hasMinimalCriteria` method in the `gw.account.AccountSearchCriteria` class.

Creating an Account

You can create an account from:

To create an account from	Path
The Account tab	Click the Account tab and select New Account .
The Desktop Actions Menu	Click Actions and select New Account .

Whichever path you select, PolicyCenter first searches to see if the account exists (name clearance). If not, it allows you to create an account.

Note: If PolicyCenter is not the system of record (SOR) for account information, you can configure PolicyCenter to synchronize account information with the SOR before creating a submission.

1. After searching for an existing account and finding none, select **Create as New Account** and then select whether the account is for a company or a person. The **Create Account** screen appears.
2. Enter the required information and select **Update**.

The **Account File Summary** screen appears, summarizing your information. The account's status is Pending, until you associate a submission with it. You can:

- Edit the account.
- Change the account holder to a new person, company, or new contact from the address book.
- Add locations, account roles, notes, documents.
- Create a submission.

Tracking Your Accounts

To track your accounts you can select **My Accounts** from the **Desktop** tab. This screen is a summary of all your pending accounts containing the account number, name, status, and address. You can further refine the search by filtering.

The default filters include:

- All Pending
- Created in Past 7 Days

Selecting an account number's link takes you to the **Account File Summary** screen.

Moving a Policy From One Account to Another

This topic describes how to move a policy from a source account to the target account.

1. Navigate to the target account.
2. From the **Actions** menu select **Move Policies to this Account**.

Note: Then menu item appears if you have the **Move policies** permission for the target account. The code for this permission is `accountmovepolicies`.

PolicyCenter displays the **Move Policies Account Selection** screen. This screen contains an account search popup for selecting the source account. The screen includes the usual account search fields and a **Related to** check box to find accounts related to this account. The search has the same minimum search criteria, validation, and security rules as other account search screens.

Use the **Related to account number** check box to restrict the search to related accounts only.

Since there is no reason to move a policy from an account to itself, the **Search Results** filters out the target account. If you attempt to search for the **Account Number** of the target account, PolicyCenter displays a warning message, and the search returns no results.

3. Click **Select** to select a source account in the **Search Results**.

PolicyCenter displays a **Move Policies Selection** popup that allows you to select one or more policies to move. The popup displays one row for each policy owned by the source account for which you have view permission. The popup includes policies even if they are canceled, expired, scheduled, or in progress policy transactions.

The **Policies** search result list view has the following columns:

Column	Description
Policy #	Click the link in this column to view the PolicyFile or JobWizard for this policy.
Policy Type	Displays the policy type.
Status	Displays the policy status as of the current date.
Policy Started	Displays the PeriodStart date of the earliest PolicyPeriod in the policy.
Current Effective Date	Displays the PeriodStart date of the latest PolicyPeriod in the policy.
Current Expiration Date	Displays the PeriodEnd date of the latest PolicyPeriod in the policy.

4. Select one or more policies, and the **Move Policies to this Account** button becomes available. When you click the **Move Policies to this Account** button, PolicyCenter calls the `movePoliciesFrom` method in `gw.account.AccountBaseEnhancement`. This method performs the following actions:

- a. Validates that all selected policies are appropriate for the target account. This validation prevents moving a personal auto policy to a company account, for example. You receive an error message if you attempt this.
- b. The `transferPolicies` method moves each policy to the target account:
 - Moves all policy terms, policy transactions, and everything else associated with the policy, including activities, notes, and documents.
 - Copies account contacts and locations referenced by the policy to the new policy.
 - Creates an activity for each moved policy.
 - Invokes the `IAccountPlugin`. See “Transferring Policies in the Account Plugin” on page 337 for more information.
- c. Generates **Policy moved** history events with a description. The method generates two history events for each moved policy, one on the source account and one on the target account. The method links the history event of the target account to the newly moved policy or policy transaction.

If the policies are moved successfully, PolicyCenter returns you to the account file of the target account. The **Policy Terms** list view and **Pending Policy Transactions** list view show the moved policies and policy transactions.

Rewriting Policies From One Account to Another

This topic describes how to rewrite one or more policies from a source account to a target account.

1. Navigate to the target account.
2. From the **Actions** menu select **Rewrite Policies to this Account**.

Note: This menu item appears if you have the **Rewrite policies to account** permission for the target account. The code for this permission is `accountrewritepolicies`.

PolicyCenter displays the **Rewrite Policies Account Selection** screen. This screen contains an account search popup for selecting the source account. The screen includes the usual account search fields. The search has the same minimum search criteria, validation, and security rules as other account search screens.

Use the **Related to account number** check box to restrict the search to related accounts only.

Since there is no reason to rewrite a policy from an account to itself, the **Search Results** filters out the target account. If you attempt to search for the **Account Number** of the target account, PolicyCenter displays a warning message, and the search returns no results.

3. Click **Select** to select a source account in the **Search Results**.

PolicyCenter displays a **Rewrite Policies Selection** popup that allows you to select one or more policies to rewrite. The popup includes canceled or expired policies for which you have view permission. The popup displays one row for each term of a policy that can be rewritten. For example, a policy has three terms. The third term was canceled flat, and the second term was canceled midterm. The popup displays both canceled terms for rewrite. However, you can rewrite only one of them.

The **Policies** search result list view has the following columns:

Column	Description
Policy #	Click the link in this column to view the PolicyFile or JobWizard for this policy.
Policy Type	Displays the policy type.
Status	Displays the policy status as of the current date.
Policy Started	Displays the PeriodStart date of the earliest PolicyPeriod in the policy.

Column	Description
Effective Date	Displays the PeriodStart date of the latest PolicyPeriod in the policy.
Expiration Date	Displays the PeriodEnd date of the latest PolicyPeriod in the policy.

4. Select one or more policies, and the **Rewrite Policies to this Account** button becomes available. When you click the **Rewrite Policies to this Account** button, PolicyCenter starts a rewrite new account policy transaction for each policy. The default effective date of each policy transaction is one of the following:

- If this is a rewrite of a canceled policy term, the default effective date is the cancellation date.
- If this is a rewrite of an expired policy, the effective date is the period end of the last term on the policy.

PolicyCenter generates an activity for each policy transaction and assigns it to the current user. The activity is a reminder to complete the policy transaction. PolicyCenter also adds a **Rewrite New Account job created** history event to the policy term of the source period.

The rewrite new account policy transaction (job) must be completed before the policy is rewritten to the new account.

Merging Accounts

This topic describes how to merge a source account to a target account.

Note: You must have the **Merge accounts** permission to view the **Actions → Merge Account into this Account** menu item. The code for this permission is `mergeaccounts`.

1. Navigate to the target account.

2. From the **Actions** menu select **Merge Account into this Account**.

PolicyCenter displays the **Select Account to Merge into Account** screen. The screen includes the usual account search fields. The search has the same minimum search criteria, validation, and security rules as other account search screens.

Use the **Related to account number** check box to restrict the search to related accounts only.

Since there is no reason to merge an account with itself, the **Search Results** filters out the target account. If you attempt to search for the **Account Number** of the target account, PolicyCenter displays a warning message, and the search returns no results.

3. Click **Select** to select a source account in the **Search Results**.

PolicyCenter displays the **Merge Account into Account** screen. This screen displays the following information about the source account:

- Account information
- Current Activities
- Policy Terms
- Pending Policy Transactions

This screen displays a message that the two accounts will be merged, and that the source account will be removed.

4. To merge the two accounts, **Click Merge Accounts**.

PolicyCenter displays a prompt asking you to confirm the merge. This prompt is to avoid accidentally removing the source account.

5. Click **OK** to merge the source account to the target account.

PolicyCenter creates a history event on the target account. The history event includes the account number of the source account.

See also

- “Overview of Merging Accounts” on page 325

Adding an Account Relationship

Note: You must have the **View account file related accounts** permission to view the **Related Accounts** screen. The code for this permission is `accountrelations`.

1. Navigate to an account and click **Related Accounts** in the left sidebar.

2. On the **Related Accounts** screen, click **Add**.

You must have the **Edit account** permission to view the **Add** button. The code for this permission is `editaccountsummary`.

3. On the **Account Relationships** screen, select a **Relationship**. In the base configuration, the choices are:

- Parent of
- Child of
- Common Ownership

Choose a **Related Account** by entering an account number or selecting the account picker icon. If you enter an account number, the number must match an existing account exactly.

4. Select the account picker icon.

5. In the **Search Related Account** screen, enter search criteria and click **Search**.

PolicyCenter displays the search popup, which contains the same search fields as the **Account Search** screen. The search results only include accounts for which you have producer code security access.

6. Click **Select** next to an account to choose it.

7. On the **Account Relationship** screen, click **Update** to add the new account relationship.

A related account appears on the **Related Accounts** screen of the related account as well.

Modifying an Account Relationship

You can modify an existing relationship by clicking the link in the **Relationship** field on the **Related Accounts** screen. PolicyCenter displays the same **Account Relationship** popup as when you add a new relationship.

PolicyCenter validates the account when you click **Update** to modify the relationship.

Removing an Account Relationship

If you select one or more rows on the **Related Accounts** screen, PolicyCenter enables the **Remove** button.

Note: You must have the **Edit account** permission to view the **Remove** button. The code for this permission is `editaccountsummary`.

Searching for Accounts with a Shared Contact

On the **Related Accounts** screen, you can search for **Accounts with a common account holder or named insured**.

If you click **Search**, PolicyCenter finds accounts with contacts that are account holders or named insureds on both accounts. The contact in common does not have to be a named insured or account holder on both accounts. For example, if a contact is an account holder on one account and a named insured in another, these are related accounts.

See also

- “Configuring Shared Contact Search Criteria” on page 341

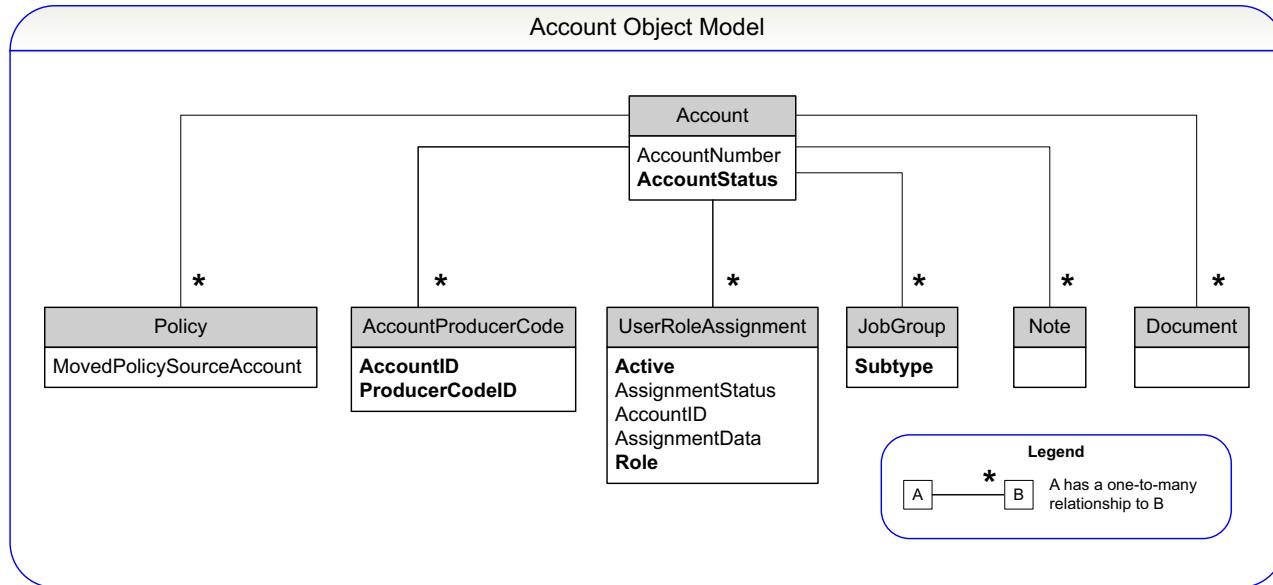
Configuring Accounts

This topic describes some of the basics of account configuration.

Account Object Model

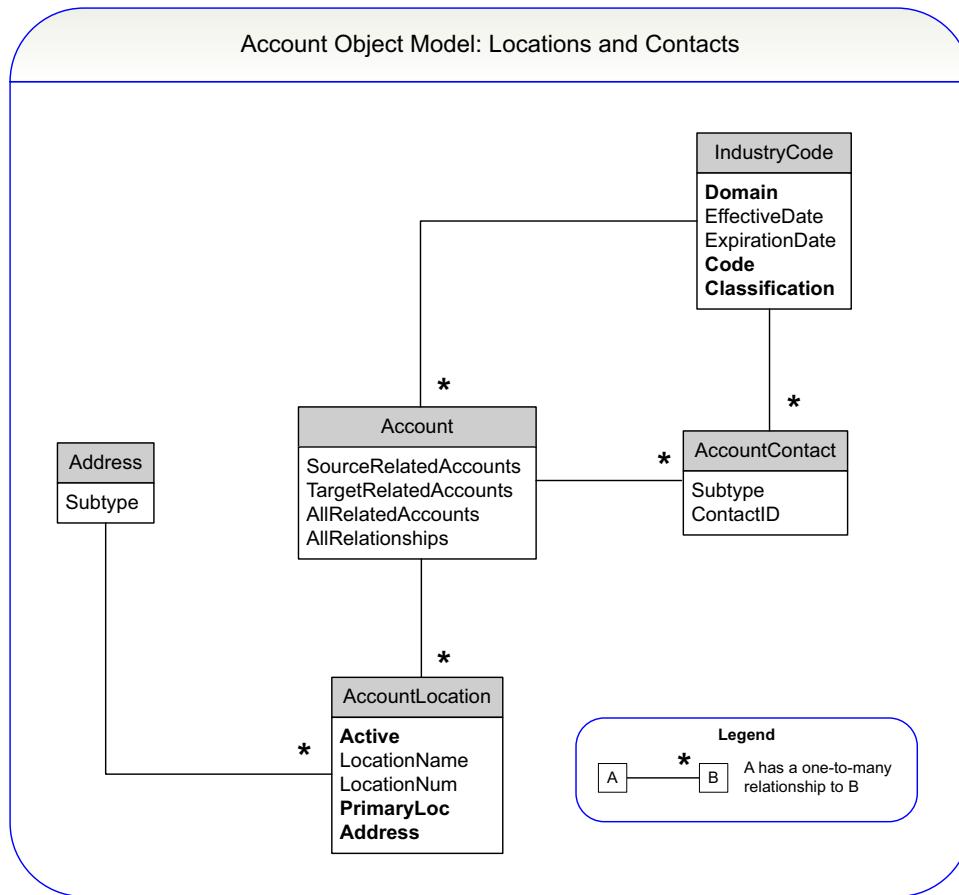
The Account object model helps you to better understand the entity relationships of accounts.

The illustration shows a partial list of the account object model. See the *Data Dictionary* for a complete list of all properties in the entities.



The Account entity contains an **AccountStatus** property. The status can be Pending, Active, or Withdrawn. You can configure the typelist for this property in Studio.

The following illustration shows account entities associated with locations and contacts.



The Address entity has the subtype AccountLocationAddress.

Account Rule Sets

In Studio, rule sets are grouped by function for the purpose of customizing a process. This topic describes the rule sets that pertain to accounts.

Note: For information on how to insert and configure Gosu rules, see “Rule Set Categories” on page 35 in the *Rules Guide*.

The following table describes rule sets that pertain to accounts in the default application.

Rule	Description
Assignment → Default Group Account Assignment Rules	Invoked when the role on the account is assigned to a group and further assignment within the group is required. Assigns users to roles on an account.
Assignment → Global Account Assignment Rules	Invoked when no group is specified as the starting point for the assignment on the account. Assigns role to a group.

Account Web Service

Use the Account API web service to create, find and manipulate accounts within PolicyCenter. External systems can use this web service to work with PolicyCenter accounts.

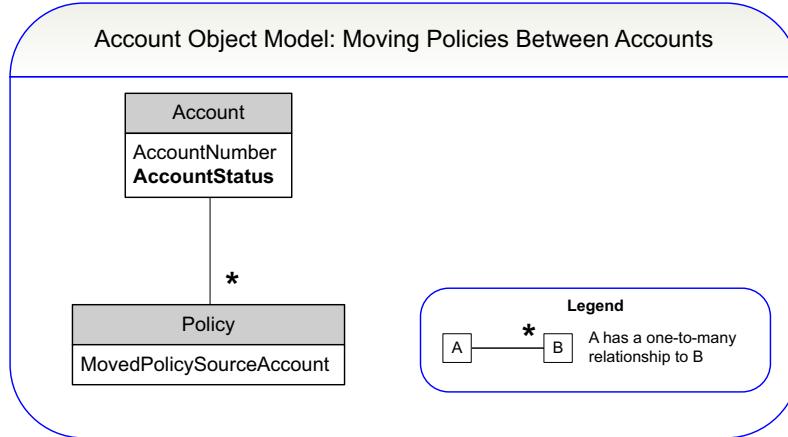
For information on how to integrate Account APIs and how to create new operations, see “Account Web Services” on page 166 in the *Integration Guide*.

Configuring Moving Policies Between Accounts

This topic describes how to configure moving policies between accounts in PolicyCenter.

Moving Policies Between Accounts Data Model

The Policy entity has a MovedPolicySourceAccount foreign key which points to the source Account. This field is null for any policy which was not moved.



See the *Data Dictionary* for a complete list of all properties in the entities. The illustration displays a partial list.

Transferring Policies in the Account Plugin

When you move policies between accounts, PolicyCenter moves the policies then calls the `transferPolicies` method of the `IAccountPlugin`. The code for this plugin is in `gw.plugin.account.impl.AccountPlugin`. In `AccountPlugin.gs`, the default implementation of the `transferPolicies` method does nothing. You can modify this method if you need to execute additional transfer logic. For example, this plugin can notify an external system about the policy move. If you have additional entities that reference accounts, you can modify those entities.

Configuring Account Relationships

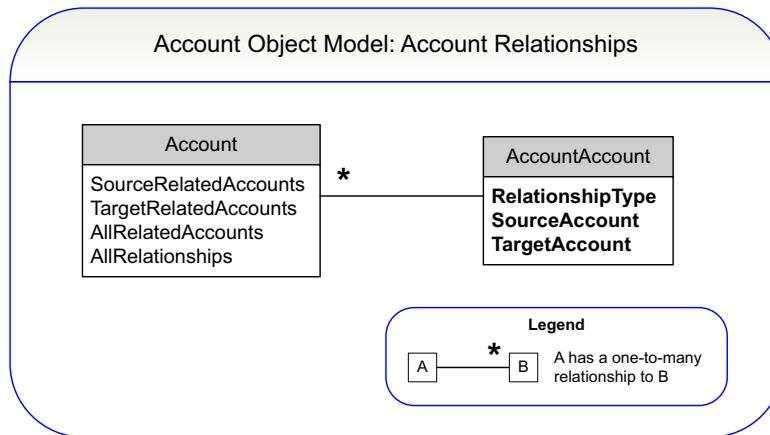
This topic describes how to configure account relationships and contains the following topics:

- “Account Relationship Object Model” on page 337
- “Account Relationship Typelist” on page 338
- “Account Relationship Methods in the Account Plugin” on page 339
- “Account Relationship Methods in Gosu Classes” on page 339

Account Relationship Object Model

The following illustration shows account entities associated with account relationships.

See the *Data Dictionary* for a complete list of all properties in the entities. The illustration displays a partial list.



The AccountAccount entity represents a relationship between a SourceAccount and a TargetAccount. The SourceAccount and TargetAccount can be the same. The RelationshipType field is a typekey to the AccountRelationshipType typelist. For more information, see “Account Relationship Typelist” on page 338.

For account relationships, the Account entity has two arrays: SourceRelatedAccounts and TargetRelatedAccounts. Each array contains AccountAccount entities, which link back to Account with the SourceAccount and TargetAccount fields. The AllRelatedAccounts field is a derived array which returns an AccountAccount entity for all source and target related accounts. For most account relationships, this field returns two AccountAccount entities. If an account is related to itself then the return array includes only one AccountAccount entity, even though that AccountAccount appears in both SourceRelatedAccounts and TargetRelatedAccounts.

A single AccountAccount entity represents a bidirectional relationship from its SourceAccount to its TargetAccount. For example, if account A is the parent of account B, then the single AccountAccount also represents that B is the child of A.

The SourceRelatedAccounts array is marked as `owner` which forces account validation to run when an AccountAccount in this array is added or updated. The default validation ensures that no duplicate relationships are created. The TargetRelatedAccounts array does not need to be `owner` because the target accounts are included implicitly in the validation rules. For more information about the `owner` attribute, see “`<array>`” on page 178 in the *Configuration Guide*. In Studio, the `owner` attribute is set in `Account.eti` located in `configuration → config → Metadata → Entity`.

The SourceAccount, TargetAccount, RelationshipType and Retired fields of each AccountAccount entity must be unique. Therefore, no SourceAccount may be related to the same TargetAccount with the same RelationshipType more than once.

Account Relationship Typelist

The AccountRelationshipType typelist is extensible, and represents the relationship from the perspective of the source account. The following table lists the codes for the AccountRelationshipType typelist in the base configuration.

Typecode	Name
parent	Parent of
child	Child of
commonowner	Common Ownership

In the base configuration, the parent and child relationships are the inverse of each other, while commonowner is reciprocal. For example, if account A is the parent of account B, then B is the child of A. Similarly, if A is a common owner with B, then B is also a common owner with A.

Account Relationship Methods in the Account Plugin

The `getInverseRelationshipType` method of the `IAccountPlugin` takes as input a relationship type, and returns the inverse relationship type. PolicyCenter assumes that all relationships have an inverse relationship, therefore this method must not return null. You can modify the default behavior of `getInverseRelationshipType` method and add code to handle new typecodes.

Account Relationship Rule Sets

In Studio, rule sets are grouped by function for the purpose of customizing a process. This topic describes the rules that pertain to account relationships.

Note: For information on how to insert and configure Gosu rules, see “Rule Set Categories” on page 35 in the *Rules Guide*.

The following table describes rule sets in the default application related to account relationships.

Rule	Description
Validation → Account Validation Rules → Related Accounts	<p>Invoked when an <code>AccountAccount</code> entity is added or updated.</p> <p>Performs the following checks:</p> <ul style="list-style-type: none">Checks that no exact duplicate <code>AccountAccount</code> entities exist on an account, with the same <code>SourceAccount</code>, <code>TargetAccount</code> and <code>RelationshipType</code> fields.Checks that no inverse duplicate <code>AccountAccount</code> entities exist on an account, where the <code>SourceAccount</code>, <code>TargetAccount</code>, and <code>RelationshipType</code> of one matches the <code>TargetAccount</code>, <code>SourceAccount</code>, and inverse <code>RelationshipType</code> of the other. For example, if A is the parent of B, then there must not be another <code>AccountAccount</code> explicitly marking B as the child of A. <p>If necessary, you can modify or add additional validations.</p>

Account Relationship Methods in Gosu Classes

This topic describes various methods in Gosu classes for working with account relationships.

Adding a Relationship

Use the `Account.addRelationship` method in `AccountBaseEnhancement` to add relationships to an account. This method creates and returns a new `AccountAccount` entity. The new `AccountAccount` appears in `Account.SourceRelatedAccounts` and in `relatedAccount.TargetRelatedAccounts`.

Removing a Relationship

You can remove a relationships by simply removing the `AccountAccount` entity in one of the following ways:

- Calling the `removeFromSourceRelatedAccounts` or `removeFromTargetRelatedAccounts` method on an `Account` entity
- Calling `remove` on the `AccountAccount` entity

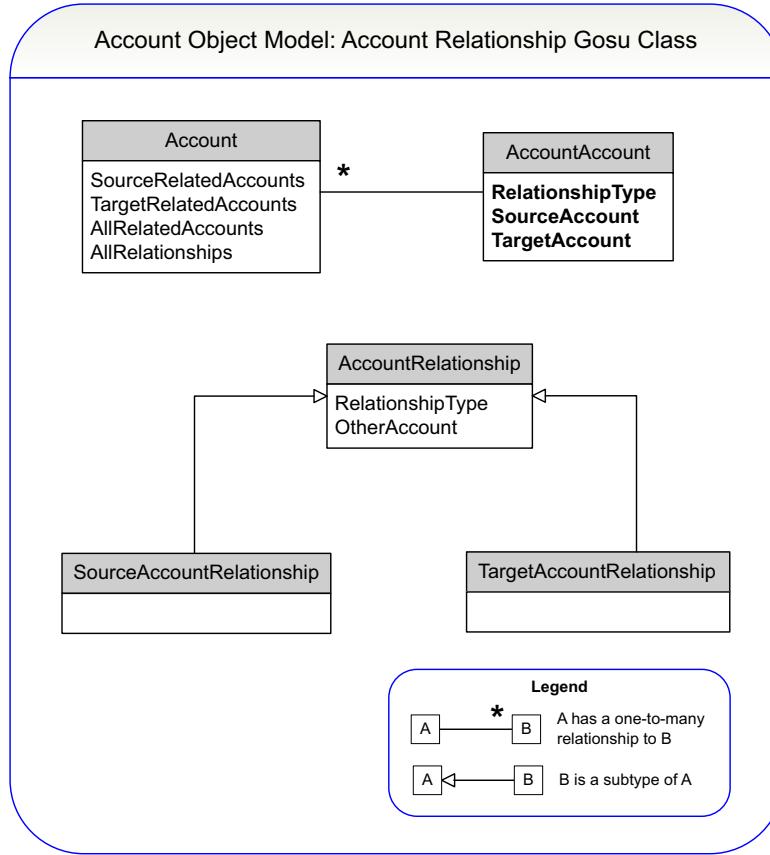
Getting Relationships

The `getRelationship` method in the `AccountAccountEnhancement` wraps an `AccountAccount` in a new `AccountRelationship` Gosu object that is aware of the direction of the relationship.

```
AccountAccount.getRelationship(primaryAccount : Account) : AccountRelationship
```

This method returns an `AccountRelationship` which represents an `AccountAccount` from the perspective of the argument `primaryAccount`. The `AccountRelationship` provides two properties, which are both readable and writable:

- `OtherAccount` refers to the other account in the relationship from the perspective of `primaryAccount`. So if `primaryAccount` is the `SourceAccount` of a relationship then `OtherAccount` maps to the `TargetAccount`. Conversely, if `primaryAccount` is the `TargetAccount` in the relationship, then `OtherAccount` maps to the `SourceAccount`.
- `RelationshipType` is the type of relationship from the perspective of `primaryAccount`. If `primaryAccount` is the `SourceAccount` of an `AccountAccount` relationship, then `RelationshipType` is the same as the type in the `AccountAccount`. If `primaryAccount` is the `TargetAccount`, then `RelationshipType` is the inverse of the type in the `AccountAccount`.



Example

Assume there is an `AccountAccount` whose `SourceAccount` is A, `TargetAccount` is B, and `RelationshipType` is parent. The `AccountAccount` object has the following relationships:

- `AccountAccount.getRelationship(A).OtherAccount == B`
- `AccountAccount.getRelationship(A).RelationshipType == "parent"`
- `AccountAccount.getRelationship(B).OtherAccount == A`
- `AccountAccount.getRelationship(B).RelationshipType == "child"`

Getting All Relationships

The `Account.AllRelationships` method in `AccountBaseEnhancement` returns an array of `AccountRelationships` for all relationships of an account.

```
Account.AllRelationships : AccountRelationship[]
```

Configuring Shared Contact Search Criteria

On the **Related Accounts** screen, you can search for **Accounts with a common account holder or named insured**. In the base configuration, the search finds accounts with contacts that are account holders or named insureds on both accounts.

You can view and modify the code for the search in `gw.account.SharedContactAccountSearchCriteria`.

The `SearchableSharedContactRoles` property defines the contact roles that this search includes.

Locations

Locations are physical places that policies define and contractually enforce. Nearly all lines of business contain information concerning locations. For example, a workers' compensation policy contains the company's main location. It also contains three additional locations for the company's warehouses. In PolicyCenter, you can create and edit locations from the account level and share them across all the policies on the account.

This topic describes how to create and maintain locations at both the account and policy levels.

This topic includes:

- “Location Overview” on page 343
- “Location Object Model” on page 346
- “Account Synchronization Classes for Locations” on page 347
- “Working with Account Locations” on page 348
- “Working with Policy Locations” on page 349
- “Configuring Locations” on page 351

Location Overview

In the insurance industry, locations are only legally enforced on the policy. Typically, you entered location information on the policy during a policy transaction, such as a submission, or a policy change, or a renewal. This process meant that if you used the same location across different policies, you had to reenter it each time. There was no linking between locations that represented the same location on different policies. In PolicyCenter, you can define and edit locations at the account level in the **Account** file in the user interface. The many benefits include consistency, ease of maintenance, less redundancy, and fewer errors. You can also create and modify location information on the policy level and have location information propagate to the account and to other non-bound policy transactions. Most location information is shared across policies. An update to the shared information propagates across all unbound usages of a location. Other information is policy or usage specific, and does not propagate to other policies.

Types of Location Information

The kinds of information that PolicyCenter stores about a location are:

- **Basic location information** such as address, city, and state.
- **Geocoding information** assigns a latitude and longitude to a location. For more information, see the following section.
- **Account-specific information** such as location name and phone number.
- **Policy-specific information** such as tax location.

Geocoding Locations

In PolicyCenter, geocoding assigns a latitude and longitude to location addresses. Geocoding location addresses lets you assemble locations into location groups by searching for nearby locations. The search for nearby locations takes into account factors such as lines of business or the status of policies and policy transactions.

See also

- “Configuring Geocoding” on page 354
- “Geographic Data Integration” on page 301 in the *Integration Guide*

Synchronizing and Revisioning Location Information

You define on a field by field basis whether to revision basic or account level information. All information at the policy level is revisioned.

While a policy transaction is open, PolicyCenter synchronizes most information in the account location with the policy location. At any time, changes can be made to an account location from the user interface of the application. If the change is made to one of the synchronized fields, then that location in any unbound policies synchronizes with the new value.

In the default configuration, PolicyCenter revisions most of the location information. Because most location information is part of the policy contract, PolicyCenter must store that information about the location exactly as it was at the time that the policy was bound. Revisioning of locations is similar to revisioning of other information on the policy. For example, the address was entered as *122 Main, Apt D* when the submission was bound. The insured calls in to report a small mistake in the address, and the address on the account is updated to *122 Main Street, Suite D*. The address on the bound policy remains unchanged. A larger mistake in the location might require a policy change which reissues the documents. When you do a policy change, the location information is synchronized from the account.

Some of the location information is not part of the policy contract and does not need to be revisioned. In the default configuration, the location name, location number, and phone number are not revisioned. For example, the phone number for a location three years ago is not important, but you need to know what it is now. The account and policy always displays the most current phone number.

Changing Location Information

When location information changes, it impacts:

- **Pending policy transactions** – The changes are immediately apparent when you view pending policy transactions because those policy transactions always display the up-to-date information. This information comes from the associated basic and account-level location information.
- **Quoted policy transactions** – When you try to bind a quoted but not bound submission, PolicyCenter verifies that the revised policy location information matches the account location information. If, after the quote, there is a change to a revisioned field on the synchronized account location, the information in this field does not match. If the locations do not match, you see a validation error. This behavior is because the change can have an effect on the quote. When you quote the policy again, the application synchronizes the location.

- **Bound policies or completed policy transactions** – Bound policies or completed policy transactions have copies of the synchronized location information at the time of binding. The associated policy revision includes this information. Each bound policy or completed policy transaction is a separate policy revision.
- **New policy transactions** – When new policy transactions on existing policies begin, the locations on those policy transactions always display the most recent location information. The location information in the revision the policy transactions are based on does not matter.

Example

The Acme account has two policies: business auto and workers' compensation. When the producer created the business auto submission, it used the location from the account. However, the producer noticed a typographical error and corrected the town/city field. Unfortunately, the producer made another mistake, and corrected the ZIP code on the account one month later. Since the business auto policy was in-force, PolicyCenter did not correct the ZIP code on the policy. A few months after these changes, Acme calls to request a workers' compensation policy. The workers' compensation submission picks up all the corrections entered on the account.

Since PolicyCenter tracks each version of the policy, PolicyCenter stores the original location information on each policy. Bound policies are legally binding. Changing the location on either the account or policy, does not change the location on other policies previously bound. However, pending policy transactions (submissions, policy changes, or renewals) always display the most current account location information.

Note: When creating a new primary location, you add a new location and set it as primary. Then you change the status of the old location to inactive. See “Adding a New Location” on page 348 for details.

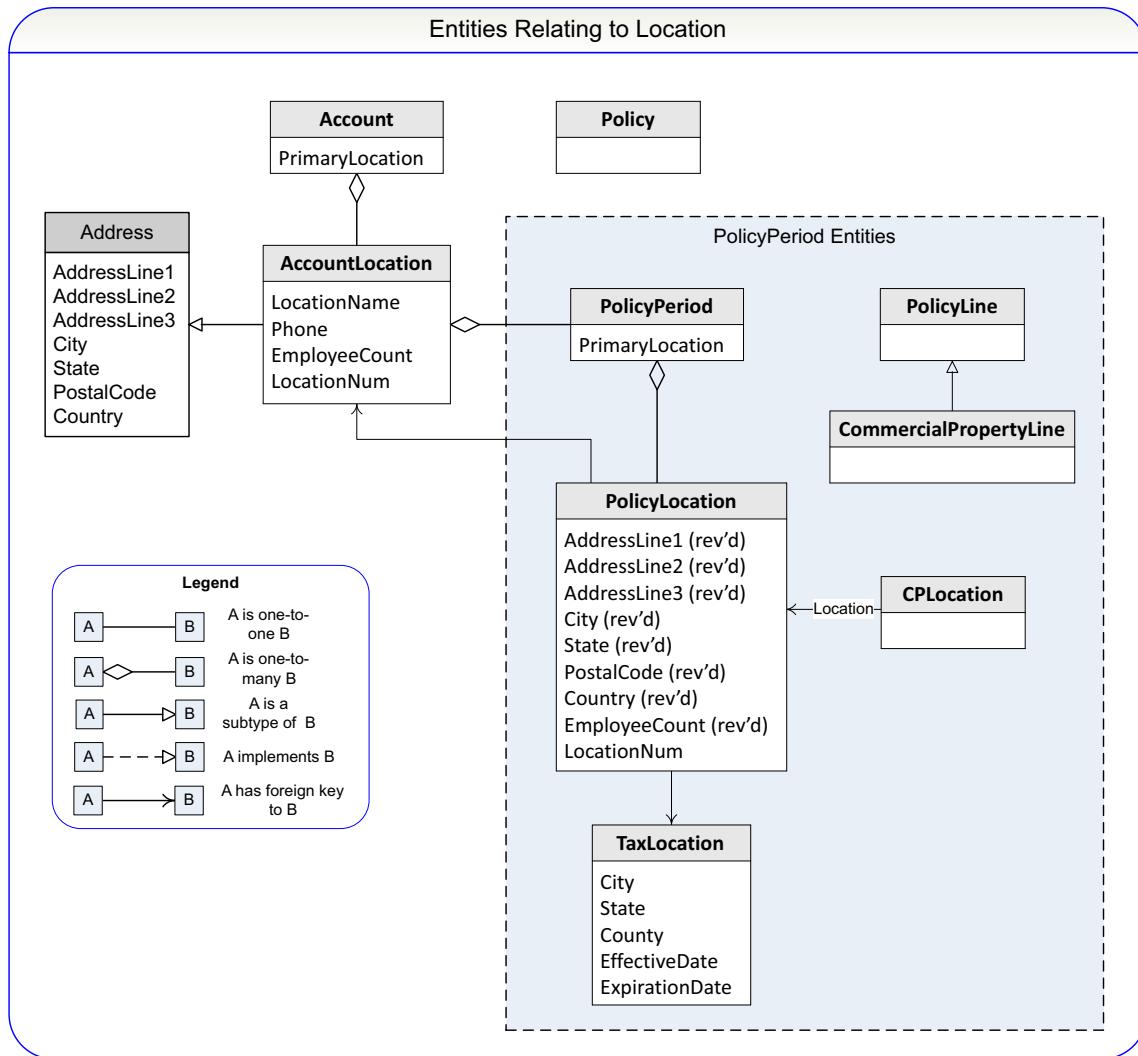
Hiding Location Information on Accounts

The default configuration of PolicyCenter has locations viewable and editable at the account level. However, some carriers may choose to only work with them at the policy level and not display them at the account level. In this case, the locations exist at the account level, even if they are not viewable. When you add locations to policies, they are implicitly added to the account.

Location Object Model

The following object model displays the key entities relating locations.

Note: See the *Data Dictionary* for a complete list of entities and properties. This illustration displays a partial list.



In PolicyCenter, you can create locations on the account level and the policy level. PolicyCenter stores location information on the account, and policies on that account can access it. The location object model is designed to handle revisioned fields. You can add fields to the **AccountLocation** entity and then configure whether those fields are revisioned at the policy level.

The **AccountLocation** entity is a subtype of the **Address** entity. You can add revisioned fields to the **AccountLocation** entity. In the default configuration, all fields in **Address** and **AccountLocation** are revisioned except for **LocationName**, **LocationNum**, and **Phone**.

Inside the box labeled **PolicyPeriod Entities**, the **PolicyLocation** entity contains fields that are revisioned. These are marked as rev'd.

The **PolicyLocation** entity contains foreign keys to the **AccountLocation** and the **PolicyPeriod** entities. The **PolicyLocation** entity has a foreign key to the **TaxLocation** entity. You can add fields to the **PolicyLocation** entity. These fields are used only on a specific policy, not across policies.

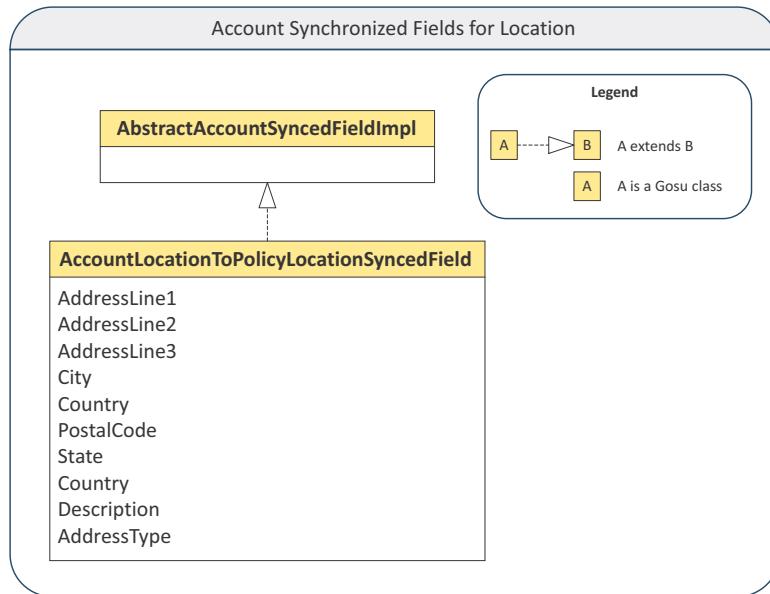
Some of the subtypes of `PolicyLine` entity have associated location types. The subtypes have a foreign key, `Location`, that points to the `PolicyLocation`. The object model diagram shows the `CommercialPropertyLine` subtype.

Subtype of <code>PolicyLine</code>	Associated Location
<code>BusinessOwnersLine</code>	<code>BOPLocations</code>
<code>CommercialPropertyLine</code>	<code>CPLocation</code>
<code>InlandMarineLine</code>	<code>IMLocation</code>

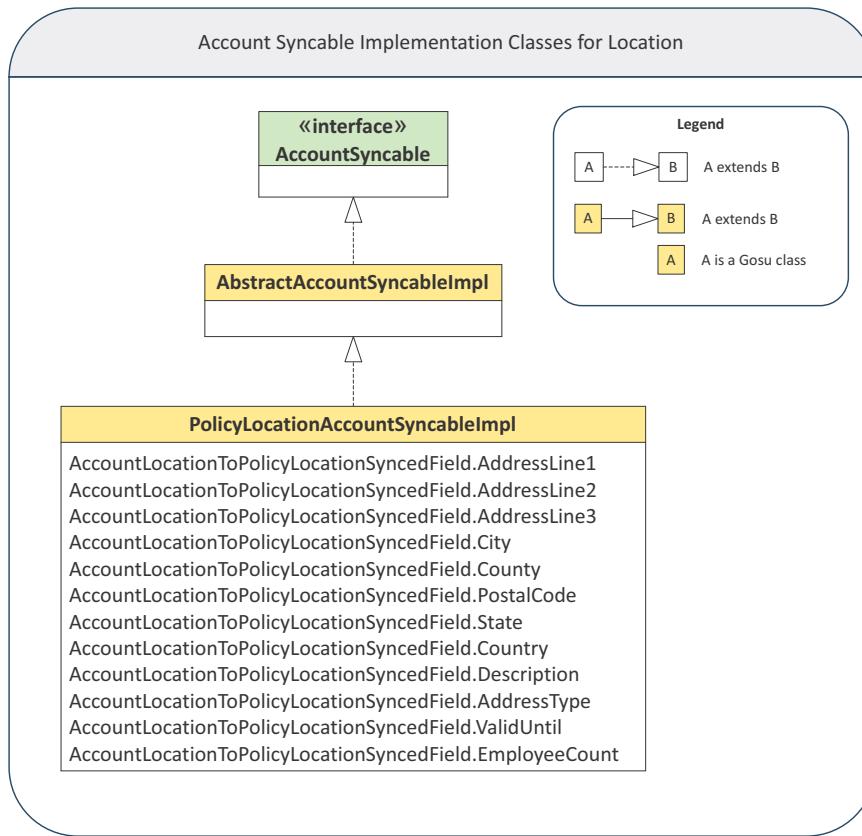
In the default configuration, account location and policy locations are numbered separately. Because of this, both `AccountLocation` and `PolicyLocation` have their own `LocationNum` field which is not revisioned. For more information about location numbering, see “Location Numbering” on page 350.

Account Synchronization Classes for Locations

The following illustration shows the Gosu classes that implement synchronized fields on the account.



The following illustration shows how the `AccountSyncable` interface is extended for locations.



Working with Account Locations

This topic explains, from a user's point of view, how to work with location information at the account level in the default configuration of PolicyCenter. For instructions on how to navigate to an account, see "Working with Accounts" on page 329.

The default account location fields are the minimum set of fields generally used across policies. Account location fields also include fields that PolicyCenter maintains at the account level for convenience although their use across policies may be limited.

The following topics assume that you already know how to access an account.

- "Adding a New Location" on page 348
- "Editing a Location" on page 349
- "Making a Location Primary" on page 349
- "Making a Location Active or Inactive" on page 349

Adding a New Location

Before adding a new location, you must already have selected or created an account. Navigate to the **Account File Summary** screen.

1. On the sidebar, click **Locations**. The **Account File Locations** screen appears.
2. Click **Add New Location**. The **Location Information** screen appears.
3. Enter your data and click **Update**. The **Account File Locations** screen appears.

Editing a Location

Edit a location only to make corrections or minor updates. Do not modify an existing location simply to create a new one. For example, if the headquarters of a business moves from *Salem, OR* to *Portland, OR*, remove the *Salem, OR* location and add a new *Portland, OR* location.

You must first find the account of the location you want to modify. Navigate to the **Account File Summary** or **Account File Locations** screen.

1. Under the **Account File Locations**, select the location number link.
2. The **Location Information** screen opens in edit mode. Make your changes and click **Update**.

Note: When editing a location, you cannot change the following fields: **Non-Specific Location**, **State**, and **Country**.

Making a Location Primary

Typically when creating an account, the account holder's address becomes the account's primary location. However, you can select another location to be primary.

Note: The primary location on the account and policy can be set to different locations. If you set a location as primary on a policy, it does not affect the account level location settings.

You must first find the account of the location you want to modify. Navigate to the **Account File Summary** or **Account File Locations** screen.

1. On the **Account File Locations** header, select the check box of the location you want to make primary.
2. Click **Set as Primary**.

Making a Location Active or Inactive

You can delete a location if no policy transaction uses it. If a location is used by a policy transaction, you can make the location inactive, making it unavailable to newly created policies. You can also, for business reasons, make a location active again so that it is available. PolicyCenter does not remove inactive locations from bound policies. This behavior is because locations only have legal meaning at the policy level, and account level locations are available primarily for usability.

You must first find the account of the location you want to modify. Navigate to the **Account File Summary** or **Account File Locations** screen.

1. On the **Account File Locations** header, select the check box of the location you want to change.
2. Click **Change Active Status**. The **Active** column shows the new status.

Working with Policy Locations

This topic explains, from a user's point of view, how to work with location information at the policy level in the default configuration of PolicyCenter.

The following topics assume that you have already created at least one account and one policy.

- “Viewing Locations on Bound Policies” on page 350
- “Editing Location Information on a Policy Transaction” on page 350
- “Adding or Removing Locations at the Policy Level” on page 350
- “Location Numbering” on page 350

Viewing Locations on Bound Policies

When viewing a bound policy, you view the location information as it was when the policy was bound. You do not see changes made to locations at the account level subsequent to binding.

In some lines of business, if the location information on the account changed after binding the policy, the **Up To Date** column on the **Locations** screen displays a **No**. Click **View** to display the **Location Details** card. If the location has changed, you see the message, **Location Information has changed since this policy was bound!** Click **View Current Location** to see the changes.

Editing Location Information on a Policy Transaction

You are able to edit locations on a policy transaction that has not been bound. You can edit an existing location or add information for a new location.

The types of fields on a policy level location screen are:

- Fields that you configure at the account level and that apply across all lines of business, such as address fields.
- Fields that you configure at the account level for ease of use but may be line of business specific, such as a **Fire Department Name**.
- Fields that you define on the line of business location (**BOPLocation** or **CPLocation**). These fields are not shared across policies.
- Fields that you define at the **PolicyLocation** level that are not used across policies but more than one line of business might use. In the default configuration, the location number is an example of this.

Changing the first two types of fields results in updating the associated account level location. Thus, they are immediately viewable on all other policy level locations on open policy transactions that reference that same account level location.

Adding or Removing Locations at the Policy Level

Adding or removing locations on a policy is a part of the job (policy transaction) wizard. However you have additional options. The options can vary based on the line of business.

- **New Location** creates a new location on the policy and on the account. The location is available to other policies under the account.
- Remove a location by selecting the locations you want to remove and clicking **Remove**. This action removes the location from the policy. The location still exists on the account.
- **Add Existing Location** allows you to add an active account location to the policy.
- **Add All Existing** adds all the active account locations to the policy.

Location Numbering

In the default configuration, account location and policy locations are numbered separately. Because of this, both **AccountLocation** and **PolicyLocation** have their own **LocationNum** field. This field is not synchronized.

The numbering of locations in the default configuration works as follows:

- If you have five locations on an account, and use locations 1, 3, and 5 in a policy, the locations are numbered 1, 2, and 3 in the policy.
- If, during a submission, you remove location 2 before binding the policy, the remaining locations are renumbered 1 and 2.
- If you have locations numbered 1, 2, and 3 in an issued policy, and remove location 2 through a policy change, then the remaining locations are numbered 1 and 3.
- At renewal, the location numbers are revised to 1 and 2.

If the default auto-numbering does not suit your needs, you can modify how the application numbers locations.

Configuring Locations

There are multiple levels of configurability.

- `PolicyLocation`, `AccountLocation`, and line of business locations are extendable, as are all of the entities in the default configuration.
- You can configure which fields are revisioned on `PolicyLocation`.
- You can configure at what points and on which policy transactions location information is synchronized.

This topic describes how to configure locations and includes:

- “Adding a Revisioned Field to a Location” on page 351
- “Creating a New Account Location” on page 354
- “Creating a New Policy Location” on page 354
- “Configuring Geocoding” on page 354

Adding a Revisioned Field to a Location

This topic describes how to add revisioned fields between account and the `Location` entity.

This topic provides general instructions and step-by-step instructions for adding a revisioned field. The step-by-step instructions guide you to add a `FloodPlain` `revisioned field` to the `AccountLocation` entity. This boolean field indicates whether the location is on a flood plain.

Follow these steps to add a revisioned field to a location:

- “Step 1: Define the Revisioned Field on the Account Location” on page 351
- “Step 2: Define the Revisioned Field on the Policy Location” on page 352
- “Step 3: Define the Field as Syncable on the Policy Location” on page 352
- “Step 4: Define the Field as Syncable on the Account Location” on page 352
- “Step 5: Add Get and Set Methods to the Policy Location” on page 353
- “Step 6: Add the Revisioned Field to PolicyCenter User Interface” on page 353

See also

- “The PolicyCenter Data Model” on page 149 in the *Configuration Guide*
- “Modifying the Base Data Model” on page 209 in the *Configuration Guide*
- “Extending a Base Configuration Entity” on page 213 in the *Configuration Guide*

Step 1: Define the Revisioned Field on the Account Location

Define the field on the account. In the default configuration for locations, the revisioned fields are on the `AccountLocation` entity. If the field already exists, you do not need to define it. If the field does not exist, add the field to an extension entity for `AccountLocation`.

The step-by-step instructions add a `FloodPlain` field to the `AccountLocation` entity.

To add the revisioned field to the account location

Define the revisioned field for `AccountLocation` in an extension entity.

1. In Studio, navigate to `configuration` → `config` → `Extensions` → `Entity`.
2. Right-click `Entity`, and select `New` → `Entity Extension`.

3. Select **AccountLocation** and click **OK**.
4. In **AccountLocation.etc** enter the following XML code to define an account level field, **FloodPlain**.

```
<?xml version="1.0"?>
<extension entityName="AccountLocation">
  <column desc="Whether the location is on a flood plain."
    name="FloodPlain" type="bit"/>
</extension>
```

Step 2: Define the Revised Field on the Policy Location

In the default configuration for locations, the revised fields are on the **PolicyLocation** entity. Define an internal field on **PolicyLocation** in an extension entity. Define the field as follows:

- Append **Internal** to the field name to signal an internal field. This convention is a Guidewire best practice for internal field names. All revised account syncable fields in the base configuration follow this naming convention.
- If this is a new field, append **_Ext** to the field name avoid name conflicts with base PolicyCenter entities.

Note: See “Defining a New Data Entity” on page 212 in the *Configuration Guide*.

- Set the value of the **setterScriptability** and **getterScriptability** attributes to **doesNotExist**. This value makes the field internal and not visible to Gosu code. The field will not appear in the data dictionary. For more information about these attributes, see “<column>” on page 180 in the *Configuration Guide*.

To define the revised field on the policy location

1. In Studio, navigate to **configuration** → **config** → **Extensions** → **Entity** and open **PolicyLocation.etc**.
2. Add a column to the entity with the following values:

Name	Value
name	FloodPlainInternal_Ext
desc	Whether the location is on a flood plain
getterScriptability	doesNotExist
setterScriptability	doesNotExist
type	bit

Step 3: Define the Field as Syncable on the Policy Location

Define the field as syncable on the policy by adding it to the **ACCOUNT_SYNCED_FIELDS** variable in the **gw.policylocation.PolicyLocationAccountSyncableImpl** Gosu class.

To define the revised field as syncable on the policy location

1. In Studio, navigate to **configuration** → **gsrc** and open **gw.policylocation.PolicyLocationAccountSyncableImpl.gs** class.
2. Add **FloodPlain** to the **ACCOUNT_SYNCED_FIELDS** variable:

```
static final var ACCOUNT_SYNCED_FIELDS = ImmutableSet.of(
  ...
  AccountLocationToPolicyLocationSyncedField.FloodPlain,
  ...
)
```

Step 4: Define the Field as Syncable on the Account Location

Add the new field to the **gw.account.AccountLocationToPolicyLocationSyncedField.gs** class. Define the new field as revised and syncable on the account.

The `AccountLocationToPolicyLocationSyncedField` constructor assumes that the field name is appended by `Internal`. If the field name is appended by `Internal`, copy one of the existing variable definitions and modify it for the new field. For example, you can copy and modify `AddressLine1`:

```
public static final var AddressLine1 : AccountLocationToPolicyLocationSyncedField<String>
    = new AccountLocationToPolicyLocationSyncedField<String>("AddressLine1")
```

If the field name is appended with `Internal` and an extension, `Internal_Ext` for example, copy one of the existing variable definitions and modify it for the new field. Then add a second parameter for the name of the field on the policy location. You must also define a constructor with two arguments for the `accountEntityFieldName` method.

The instructions describe how to define the field if it is appended by `Internal_Ext`.

To define the revised field appended with Internal_Ext

1. In Studio, open `gw.account.AccountLocationToPolicyLocationSyncedField.gs`.

2. Add the `FloodPlain` field to the `AccountLocationToPolicyLocationSyncedField` class:

```
class AccountLocationToPolicyLocationSyncedField<T> extends
    AbstractAccountSyncedFieldImpl<PolicyLocation, T> {
    ...
    public static final var FloodPlain : AccountLocationToPolicyLocationSyncedField<String> =
        new AccountLocationToPolicyLocationSyncedField<String>("FloodPlain",
            "FloodPlainInternal_Ext")
    ...
}
```

3. Define a constructor for `accountEntityFieldName` with two arguments:

```
construct(accountEntityFieldNameArg : String, policyEntityFieldNameArg : String) {
    super(accountEntityFieldNameArg, policyEntityFieldNameArg)
}
```

Step 5: Add Get and Set Methods to the Policy Location

The `gw.policylocation.PolicyLocationEnhancement` has get and set methods for syncable fields. To find these methods, open the file and search for `Shared` and `revisioned`. You can copy one of the existing methods and modify the code for your field.

To add get and set methods to the policy location

1. In Studio, open `gw.policylocation.PolicyLocationEnhancement.gsx`.

2. Search for “`Shared` and `revisioned`”.

3. Copy the get and set methods for `AddressLine1`, and paste it in the file.

4. Modify the pasted code. The get and set methods for `FloodPlain` are:

```
/**
 * Shared and revisioned flood plain.
 */
property get FloodPlain() : String {
    return AccountLocationToPolicyLocationSyncedField.FloodPlain.getValue(this)
}

/**
 * Shared and revisioned flood plain.
 */
property set FloodPlain(arg : String) {
    AccountLocationToPolicyLocationSyncedField.FloodPlain.setValue(this, arg)
}
```

Step 6: Add the Revisioned Field to PolicyCenter User Interface

To see this field in PolicyCenter, you have to add it to the `PolicyFileLocations`, `AccountLocationPopup`, `AccountLocationDetailInputSet`, and other PCF pages.

Creating a New Account Location

To add a new `AccountLocation` to the account, call the `newLocation` method in the `Account` class.

Creating a New Policy Location

If the associated account location is not yet defined on the account:

- A new account level location is created on the account, and a new associated policy level location is created on the policy.
- The policy level location has a foreign key to the account level location.

If the associated account location is already defined on the account:

- A new associated policy level location is created on the policy.

To add a new `PolicyLocation` from a policy period, call one of the methods in the `PolicyPeriod` class:

- `newLocation()` – Creates a new `PolicyLocation` and a new `AccountLocation`.
- `newLocation(AccountLocation accLocation)` – Creates a new `PolicyLocation` linked to the existing `AccountLocation` passed into the method.

Configuring Geocoding

PolicyCenter provides geocoding, which assigns a latitude and longitude to an address, for locations. The default configuration provides Gosu methods to search for locations centered around an address. You can extend the search criteria to meet your business needs.

The `Address` entity contains fields for geocoding. The geocoding fields on `Address` are `Latitude`, `Longitude`, and `GeocodeStatus`. Successful geocode statuses include `exact`, `postalcode`, `street`, and `city`. Each status indicates the degree of precision in identifying the geographic location of an address.

Geocoding is performed when you create or edit an `AccountLocation`. There is also a `GeocodeWriter` batch process which runs periodically to update geocoding information on locations.

Geocoding is implemented with the `GeocodePlugin`. In the default installation, the `GeocodePlugin` is implemented by a `BingMapsPlugin` class, which integrates with the Microsoft Bing Maps Geocode Service. You must obtain a valid Bing Maps application key from Microsoft to use this `GeocodePlugin` implementation.

See also

- “Using the Geocoding Feature” on page 16 in the *System Administration Guide*
- “List of Batch Processes and Distributable Work Queues” on page 97 in the *System Administration Guide*
- “Geographic Data Integration” on page 301 in the *Integration Guide*

Account Location

Account location geocoding methods are located in the `gw.accountlocation` package.

- `AccountLocationFinder.gs` provides methods for finding account locations within a specified radius. You can also find a specified number of the closest account locations. The center of these searches can be a geocoded address or latitude and longitude.

Policy Location

Policy location geocoding methods are located in the `gw.policylocation` package.

- `PolicyLocationAddressExtractor.gs` extracts the account location from a policy location.
- `PolicyLocationSearchCriteriaEnhancement.gsx` specifies search criteria and allows you to search for locations matching those criteria. This enhancement allows you to search for locations in policies:

- By line of business
- In the same account
- In bound policies
- In open submission policy transactions
- Within a specified radius

Generic Schedules

Schedules are lists that contain detailed information about an insured's coverables. PolicyCenter implements two distinct types of schedules:

- Schedules that capture information per scheduled item. Schedules of this type are not used directly for rating, but are often taken into consideration during underwriting and usually are included in forms.
- Schedules that include coverage terms per scheduled item. In schedules of this type, each scheduled item's coverage terms are passed to the rating engine and potentially affect the cost of the policy.

Generic schedules are instances of schedules that are implemented by using the generic schedule data model in PolicyCenter.

This topic provides basic information about generic schedules.

This topic includes:

- “Generic Schedule Overview” on page 357
- “Types of Schedules” on page 358

See also

- “Configuring Generic Schedules” on page 41 in the *Product Model Guide*

Generic Schedule Overview

In insurance terminology, schedules are lists of items that define, extend, restrict, or exclude coverages or coverage terms. The user interface for a schedule appears when you add or select the coverage, exclusion, or condition that requires a schedule. The user interface for a schedule is a table containing one row per scheduled item and as many columns as needed to capture the information pertaining to the scheduled item. **Add** and **Remove**

buttons manage the schedule enabling it to contain any number of scheduled items, and to change over the life of the policy.

* Description	* Type	* Bodily Injury Excl	* Property Damage Excl

Scheduled coverage summaries also appear in the **Policy Review** screen, including the **Differences** card view. The data model for generic schedules enables you to configure schedules using a common set of Gosu classes, PCF pages, and entity interfaces.

Examples of generic schedules can be found in the base product general liability line. Generic schedule examples include:

- **General Liability Coverages** – Pesticide Or Herbicide Applicator Coverage (`GLPestHerbicideApplicatorSchedule`)
- **General Liability Exclusions** – Exclude Y2K Computer Related And Other Electronic Problems - Excl of Specd Covs For Designate Products Or Completed Ops (`ExcludeY2KCompAndElecProbSchedule`)
- **General Liability Conditions** – Amendment of Section V - Extended Reporting Periods For Specific Accidents (`AmendExtRepPerdSpecAccidSchedule`)
- **General Liability Conditions** – Amendment of Section V - Extended Reporting Periods For Specific Locations (`AmendExtRepPerdSpecLocSchedule`)
- **General Liability Conditions** – Amendment of Section V - Extended Reporting Periods For Specific Products or Work (`AmendExtRepPerdSpecProdWorkSchedule`)

Schedules that have historically been a part of base product lines other than General Liability do not use the generic schedule data model. Instead, these schedules are individually implemented using dedicated entities that were defined for each schedule type. Following is an example of an individually-implemented schedule:

- **Inland Marine Coverages** – Contractor's Equipment is backed by a dedicated `ContractorsEquipment` entity.

Guidewire recommends that you add schedules to product lines using the generic schedule data model.

Types of Schedules

Although schedules can contain many types of information, PolicyCenter supports two broad types of schedules:

- **Schedules of items without coverage terms** can contain any number of columns. None of the captured data is used in rating the policy.
- **Schedules of items with coverage terms** can contain any number of columns. One or more of the columns capture coverage term data that is used to rate the policy. Examples of coverage terms include limit and deductible.

The general liability line has examples of schedules without coverage terms, but no examples of schedules with coverage terms.

Activities

In PolicyCenter, you may accomplish many tasks such as:

- Obtaining a credit or motor vehicle report before approving a submission
- Meeting with the insured to verify coverages
- Reviewing a submission before approving and issuing the policy
- Reviewing evaluation issues prior to a policy being renewed

These tasks are *activities* and PolicyCenter tracks these activities. Tracking work by using activities make it easier for you to perform all necessary policy-handling tasks and to identify missed tasks. Activities allow supervisors and managers to track assigned work and to identify policy issues such as those with many overdue or escalated activities.

This topic includes:

- “Activities Overview” on page 361
- “Working with Activities” on page 363
- “Activity Patterns” on page 364
- “Activity Object Model” on page 367
- “Activity Batch Process” on page 367

Activities Overview

In PolicyCenter, there can be many tasks that need to occur before a policy transaction can finish. For example, before issuing a business auto submission, a producer might need to obtain driver information from the Department of Motor Vehicles. An underwriter may need to review a submission that has a high level of risk before issuance. More than one user may perform these tasks, and users may handle these tasks at different times. In PolicyCenter, these tasks are associated with an account, policy, or a policy transaction. PolicyCenter tracks these activities until they are completed. If you view a **Policy File** screen, you see all the open activities associated with policy transactions on that policy. If you view an **Account File** screen, you see all the open activities associated with policy transactions or policies on that account.

Activities at the account level can include:

- Meeting with the producer.
- Reviewing the account at renewal time to see if new policies might benefit the account holder.

Activities at the policy level can include:

- Creating an activity on the policy to request motor vehicle reports for a commercial auto policy every six months to check for high risk drivers.
- Creating an activity to order loss reports for the policy four times a year. This activity can affect whether the carrier decides to renew the policy.
- Creating an activity to schedule a meeting with the underwriter to discuss pre-renewal directions.
- Creating an activity to inspect the insured's property to verify that they are properly safeguarding the property against risk.
- Stat reporting errors can generate follow-up activities requesting corrections on the policy.

Activities at the policy transaction level can include:

- Referring a submission to an underwriter for approval.
- Gathering information or reports so that a final audit can be completed.
- Following up on activities after a submission policy transaction, such as getting a property inspection report, getting signatures from the insured, and so forth.

Activities Creation and Assignment

You can generate and assign an activity either manually or automatically:

- **Manually** – Through the user interface. You usually do this to request that work be done by other users or to remind yourself to do the work. For example, you manually create an activity to gather additional information before a submission can be quoted. You can then explicitly select an owner for that activity or have PolicyCenter assign it for you.
- **Automatically** – Through business rules defined in Gosu. Some examples include:
 - A business rule determines that an underwriter needs to review a producer's submission before moving it forward.
 - An activity is past due, and PolicyCenter creates an activity to alert the supervisor.

See also

- “Creating an Activity” on page 363 for information on how to manually create an activity.
- “Assigning Activities” on page 363 for information on how to assign one or more activities.
- “Assignment in PolicyCenter” on page 55 in the *Rules Guide* for information on how PolicyCenter assigns an activity.
- “Defining Activity Patterns” on page 403 in the *Configuration Guide* for information on how to automatically create an activity.

Activity Ownership

How do you know if you have activities assigned to you? When you are assigned an activity, it appears in your My Activities desktop. Select an activity to view details and to take the appropriate action. Only you can edit that activity unless others, such as your supervisor, have permissions. Optionally, you can reassign an activity.

Activities are assigned to a user directly – for example, producers might assign activities to themselves. Activities are also assigned by role based on routing rules. Changes to any policy transaction or account does not affect the ownership of the activity.

Activity Escalation

If an activity is not worked on by a target date, the activity can be overdue or escalated. In the default application, the only indication of overdue or escalated activities is the way in which they appear in activity lists. You can configure PolicyCenter to add special functionality for overdue or escalated activities. For example, you can configure PolicyCenter to automatically reassign escalated activities to a supervisor.

Localizing Activity Patterns

You can localize the subject and description of an activity pattern through the PolicyCenter interface if you configure PolicyCenter for multiple locales.

See also

- “Pattern Types and Categories” on page 404 in the *Configuration Guide*

Working with Activities

The topics in this section explain how to work with activities.

- “Creating an Activity” on page 363
- “Assigning Activities” on page 363
- “Completing or Skipping Activities” on page 364
- “Selecting an Activity from a Queue” on page 364

Creating an Activity

- To create an activity, first navigate to the object that you want to attach the activity to. You can attach activities to policy transactions, account, and policies. In this example, the activity is to verify coverage in a policy change.
- Select **New Activity** from the **Actions** menu. Select the category (**Reminder** in this example) and the type of activity (**Verify Coverage**).
- Enter the required information. You can either select a person to handle the assignment or have PolicyCenter assign it for you.
- Optionally, you can add a **New Note** at the same time.
- Click **Update**. The activity owner can view the new activity on their **Desktop** under **My Activities**. Anyone who has permissions to view an account under the **Account File Summary** screen can also view the activity.

Note: You can view activities on other screens in PolicyCenter. You can view activities on the **Workplan** for the policy transaction, and you can view all open policy transaction activities on the **Policy File** screen.

Assigning Activities

You can reassign an activity that you own:

- Navigate to the activity and click **Assign**.
- Navigate to a list containing the activity.
Select the check box to the left of one or more activities.
Click **Assign**.

Doing either displays the **Assign Activities** screen where you can assign the activity through assignment, or specify a user, group, or queue.

Completing or Skipping Activities

You can update, skip, or complete an activity.

1. Go to **Desktop** → **My Activities** and click an activity link in the **Subject** column.

The **Activity Detail** screen appears in the lower pane of the user interface.

If you have the **View notes** permission, there is a **View Notes** button. Click this button to view all notes attached to the current activity. The code for the **View notes** permission is `noteview`.

2. If you have the correct permissions to edit the activity, then make your changes and click **Update**.

3. After you review an activity, you can click **Skip** or **Complete**.

Skipping an activity indicates that you no longer want to do the activity. Completing an activity marks it as finished. You can also skip or complete one or more activities from the **My Activities** screen by selecting a number of them, then clicking **Skip** or **Complete**.

Selecting an Activity from a Queue

A queue is a repository which contains activities assigned to a group but not to a particular user in that group. Users in that group can take ownership of activities. If you navigate to **My Queues**, and select from the drop-down menu, you can see activities in the queue. To assign an activity to yourself, click **Assign Next to Me**.

Activity Patterns

Activity patterns are templates that standardize the way PolicyCenter generates activities. Both Gosu classes and the user interface create activities based on these patterns. Each pattern describes one kind of activity for handling the policy or account process.

Activity patterns contain many default, or typical, characteristics for each activity, such as its name, its relative priority, and whether or not it is mandatory. When an activity is created either by you or through Gosu, PolicyCenter uses the pattern as a template to set the activity's default values, such as **Subject** and **Priority**. Defaults can be overridden.

Activity Pattern Components

The following is an explanation of the various fields used in creating or modifying an activity pattern. You can do this from the **Activity Patterns** menu link in the **Administration** tab.

Field	Description
Short Subject	Enter a brief description of the activity. Use Short Subject in small areas of the user interface where Subject may be too long.
Automated Only	Required. Click Yes or No to indicate whether this pattern is not available for manually created activities and used only by rules and Gosu.
Description	Enter a description of what is expected in the completion of this activity. This field is visible only when looking at the details of the activity.
Escalation Days	Enter how many days before an activity will be escalated if not complete. Escalation also depends upon Escalation Start Point .
Escalation Hours	Enter how many hours before an activity will be escalated if not complete. Escalation also depends upon Escalation Start Point . If you specify both Escalation Days and Escalation Hours , the activity will be escalated in that many days and hours. For example, if you specify 2 escalation days and 8 escalation hours, the activity will be escalated in 56 hours ((2 x 24) + 8).
Mandatory	Required. Click Yes or No to indicate whether you can skip this activity.

Field	Description
Code	Enter the internal name for the pattern. Business rules and Gosu use this code when creating an activity or checking to see which pattern an activity was created from.
Recurring	Required. Click Yes or No to indicate whether the activity recurs on a regular schedule. When you complete a recurring activity, you click a Complete and Create New button rather than Complete button. This action automatically creates a new activity.
Subject	Required. Enter the title of the activity pattern.
Target Days	Enter the target number of days to complete this activity. The number of days also depends upon Target Start Point.
Target Hours	Enter the target number of hours to complete this activity. This number of hours also depends upon Target Start Point. If you specify both Target Days and Target Hours, the activity is targeted to be completed in that many days and hours. For example, if you specify 2 target days and 8 target hours, the activity is targeted to be completed in 56 hours ((2 x 24) + 8).
Category	Select the category of the activity. The category determines where the activity pattern appears in the New Activity screen action menu. The user interface displays related groups of patterns making it easier for you to select a pattern. The Category field classifies patterns into related groups. Each typecode of the ActivityCategory typelist is an activity pattern Category, and relates each Category to a Type. The categories in the default application are: <ul style="list-style-type: none">• Approval• Correspondence• Interview• New mail• Reminder• Request• Response• Approval denied• General• Underwriter Review
Activity Class	Required. Select whether the activity is a task or event. A task can have a due date but an event cannot.
Escalation Start Point	Select when to begin calculating the escalation date or time. Choices include: Activity creation date, Policy Effective Date, and Policy Expiration Date.
Target Start Point	Select when to begin calculating the target date. Choices include: Activity creation date, Policy Effective Date, and Policy Expiration Date.
Priority	Select from the drop-down menu whether the activity is Urgent, High, Normal, or Low.
Type	Required. Select whether the activity is a general activity or an approval activity. A general activity is closed by being completed or skipped. An approval activity is closed by being approved or denied. The default application contains a number of internal activity pattern types. Activity patterns with types other than General can be accessed by Gosu and PolicyCenter code. Do not remove these activity patterns. However, administrators can customize attributes of these internal activity patterns, such as their due dates. The activity pattern types in the default application besides General are: <ul style="list-style-type: none">• Approval – Activities to approve, such as approving a submission.• Approval Denied – Activities for reviewing a denied approval request.• Assignment Review – Assignment activities added to an underwriter's Pending Assignment queue.• Policy Approval – Activities which approve a policy.
Pattern Level	Required. Select All, Account, Job, or Policy. These indicates the levels at which the activity can be attached.

Creating and Editing Activity Patterns

Administrators or managers with sufficient permissions can view, create, and edit patterns on the **Activity Patterns** page under the **Administration** tab.

An activity pattern and an activity created from the pattern can have the same name. The pattern suggests its own name as the default activity name. You can think of a pattern as an entity, and the corresponding activity as an instance of it.

Create new activity patterns or modify existing ones by clicking the **Activity Patterns** sidebar in the **Administration** tab.

To create a new pattern

1. Click **New Activity Pattern** on the **Activity Patterns** screen.
2. Enter the data as described in “Activity Pattern Components” on page 364.
3. To save, click **Update**.

To modify an existing pattern

1. Select the desired pattern.
2. Enter the data as described in “Activity Pattern Components” on page 364.
3. To save, click **Update**.

Using Gosu to Edit Patterns

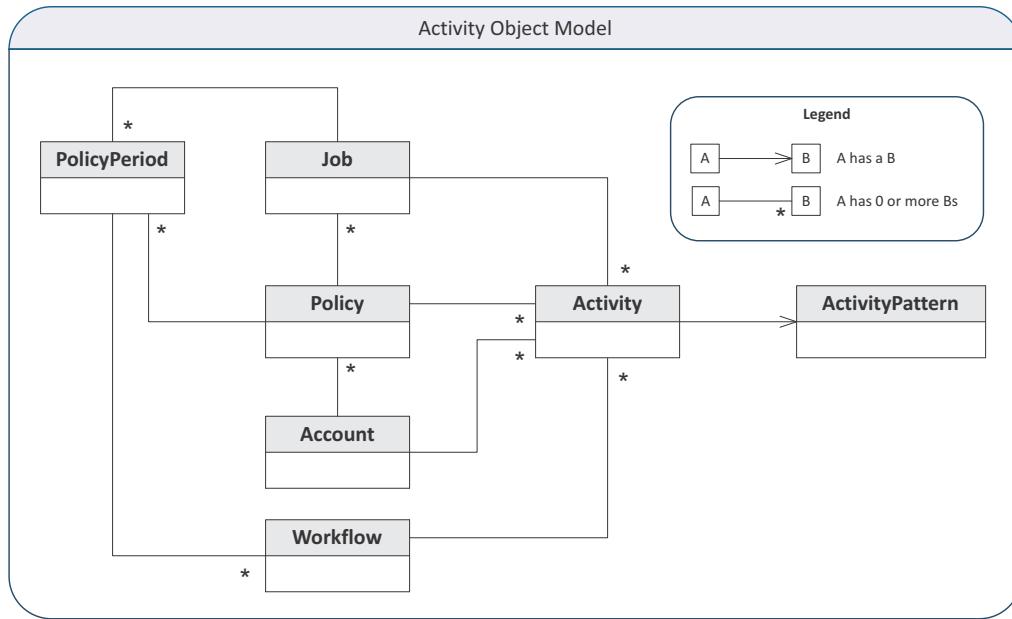
You can also create activity patterns by using Gosu in Studio.

See also

- “Pattern Types and Categories” on page 404 in the *Configuration Guide*
- “Defining Activity Patterns” on page 403 in the *Configuration Guide*

Activity Object Model

The following object model diagram describes key entities relating to Activity. For complete information, see the *Data Dictionary*.



The following table describes the entities related to Activity.

Entity	Description
Account	There can be many activities associated with an account.
Activity	The main entity which is associated with pre-defined job (policy transaction) processes and rules.
ActivityPattern	The template used to create activities. See “Activity Patterns” on page 364 for more information.
AssignedByUser	The person who assigned the activity.
Job	Many activities can be associated with a job.
Policy	Many activities can be associated with a policy.
PolicyPeriod	The policy period for the activity. This policy period is the policy period from an associated workflow. If there is none, then it comes from the associated job. If there is no associated job, it is null.
Workflow	Many activities can be associated with a workflow.

Activity Batch Process

The ActivityRetire batch process retires activities that have been canceled or dismissed.

See also

- “List of Batch Processes and Distributable Work Queues” on page 97 in the *System Administration Guide*

PolicyCenter Contacts

PolicyCenter stores contact information on policies and accounts. You can manage, group, and reuse contact information. You define and maintain contacts at the account level and use them across policies. You can have policy specific contact role information added at the policy level. You can also enter and edit contact information on a policy, and have it update the account and unbound policies in the account.

In PolicyCenter, managing contacts is similar to how you manage locations. For more information about locations, see “Locations” on page 343.

This topic includes:

- “Contact Overview” on page 369
- “Working with the Contact Tab” on page 374
- “Working with Contacts in Policies and Accounts” on page 382
- “Contact Object Model” on page 390
- “Account Synchronization Classes for Contacts” on page 393
- “Configuring Contacts” on page 394

Contact Overview

PolicyCenter defines a contact as either a person or a company. A contact exists outside of any role it happens to play. Generally, PolicyCenter requires that you enter standard data for a contact regardless of the role that the contact is assigned to. After defining a contact, you can add additional roles to it.

A contact that is set up from the **Account File** can be used by all policies within the account. One contact can play multiple roles on the account and on the policy. Take a personal auto policy for example. A contact can be the holder of the account, the primary named insured on the policy, and a driver of a vehicle insured by the policy.

You can access contacts through accounts and policies, which provide a centralized view of all contacts on the account and policy files. Some contact information is shared across policies. An update to the shared information propagates across all unbound usages of a contact. Other information is policy or usage specific, and does not propagate to other policies.

You can also access contact through the **Contact** tab on the tab bar.

The benefits of sharing contacts between accounts and policies include:

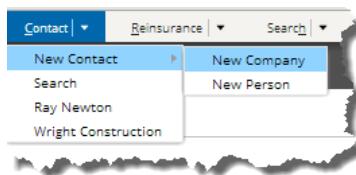
- Avoiding data reentry and errors
- Allowing the same contact to play multiple roles on the account and policy, such as account holder, named insured, or billing contact
- Allowing you to configure which pieces of contact information to revision
- Associating **Contact** to other entities such as **Location** or **Vehicles**

Centralized View of Contacts on the Contact Tab

The **Contact** tab provides a central place to view information associated with a contact such as:

- Details including name, phone, date of birth, addresses, and other information
- Accounts
- Policies
- Work orders
- Claims if PolicyCenter is integrated with claims system
- Billing if PolicyCenter is integrated with a billing system

Using the **Contact** tab, you can create new contacts, search for existing contacts, or select a recently viewed contact. You can also create an account for the contact.



See also

- “Working with the Contact Tab” on page 374
- “Configuring the Contact Tab” on page 408

Sharing Contacts with a Contact Management System

A contact management system maintains contacts in a central location. These contacts can be shared across applications. The default configuration of PolicyCenter includes an integration with Guidewire ContactManager. You can also integrate PolicyCenter with the contact management system of your choice. In the default configuration, the integration with ContactManager is not enabled.

See also

- “Contact Management System Integration” on page 721
- “Adding a Contact From the Address Book” on page 383
- “Integrating ContactManager with Guidewire Core Applications” on page 43 in the *Contact Management Guide*

Contacts and Roles

In PolicyCenter, one contact can have multiple roles. In some cases, an edit to the contact is synchronized or propagated across the contact and its roles. In other cases the change affects only the particular role. For

example, Joe is the account holder (in the account role), billing contact and secondary contact. If you edit the address for Joe, then you generally expect to see the change in all of his roles.

In addition to standard contact information, some contacts have role information that needs to be collected. For example, a contact that is in a personal auto driver role needs to provide the year first licensed. A role does not need to have role specific information to be a defined role. For example, a billing contact only requires standard contact information.

A contact can play multiple roles. For example, a contact on a personal auto policy can have the roles of named insured and driver on the policy, as well as account holder on the associated account.

See also

- “[Revisioning Contact Information in Policies](#)” on page 371

Types of Contact Information

The kinds of information that PolicyCenter stores about a contact are:

- **Contact information that is used across roles**, such as address and phone information.
- **Role-specific information on the account level** that is consistent everywhere that a contact plays that role. Some examples would be year first licensed for a driver role or an industry code for a named insured role.
- **Role-specific information on the policy level** that could be different anywhere the contact plays the role such as a driver percentage on any given vehicle for the driver’s role.

Account Holder Information

The **Account Holder Info** screen provides a summary of a PolicyCenter account holder. This summary screen consolidates information retrieved from PolicyCenter, as well as information requested from BillingCenter, ClaimCenter, and ContactManager, or equivalent third-party systems of record. This screen provides a snapshot of the account holder’s value to the organization.

The **Account Holder Info** screen is an example of a summary screen which consolidates information about an account holder from PolicyCenter and other applications. You can modify the **Account Holder Info** screen or add similar screens that display information for other account contact roles such as a **Secondary Account Holder** role.

In the default configuration, the **Account Holder Info** screen works with a complete installation of Guidewire InsuranceSuite. Therefore, some of the fields in the **Account Holder Info** screen contain meaningful data only if BillingCenter, ClaimCenter, and ContactManager are integrated with PolicyCenter. Through configuration, you can also retrieve information from third-party billing, claims, and contact systems of record. To retrieve information from third-party integrations, you must configure the **Account Holder Info** screen to display equivalent values.

The **Account Holder Info** screen retrieves information from other InsuranceSuite applications of the same release version. With additional configuration, you can make the screen with other versions of these Guidewire products, as well as with third-party billing, claims, and contact management systems.

See also

- “[Viewing the Account Holder Info Screen](#)” on page 380
- “[Configuring the Account Holder Info Screen](#)” on page 477 in the *Configuration Guide*

Revisioning Contact Information in Policies

In policies, some contact data is revisioned. All revisioned contact information is effective dated based on the edit effective date of a particular policy transaction. This revisioning behavior is the same as for all other revisioned policy data. Revisioned contact fields are accurate as of a particular point in time.

In policies, PolicyCenter revisions pieces of contact information for certain contact roles. Because certain pieces of contact information are part of the policy contract, PolicyCenter must store that information on the bound

submission exactly as it was at the time of binding. Each bound policy or completed policy transaction is a separate revision. Thus, a change to contact or contact role information has no impact on bound policies or completed policy transactions. This behavior is similar to how PolicyCenter revisions all other information about the policy. For example, when binding a submission, the name of the insured was Maria Smith. The name of the insured on the bound submission or issued policy, remains Maria Smith even if she subsequently changes her name to Maria Jones.

Information associated with contacts that is not part of the policy contract does not need to be revised. For example, the phone number for a billing contact three years ago is not part of the policy contract. You always want to see the most current phone number and do not care about past phone numbers.

For policy contacts, you define on a field by field basis for each role whether to revision basic contact information or account level role-specific information. Role-specific information on the policy level is by definition revised because everything at the policy level is by definition revised.

Depending on how the effective date of the policy transaction relates to the last update time of the contact, updates to contact information within a policy transaction can:

- Be considered current.
- Be back-dated if the effective date of a policy transaction is earlier than the last update time the contact.
- Be future-dated if the effective date of a policy transaction is later than the current date.

PolicyCenter tracks the last update time of the contact so that the policy correctly represents the contact data.

Note: Accounts are not revised, and contact data is not revised in accounts. Therefore, when you view an account in PolicyCenter, all account data, including the contact data, is the current data.

See also

- “Policy Revisioning” on page 479

Contact Revisioning When Contacts Are Synchronized

When contacts are synchronized with the account, PolicyCenter:

- Copies down contact information from the account at the start of the policy transaction.
- Copies a change to the policy contact up to the account contact throughout the steps the policy transaction.
- Copies a change to the account contact down to the policy contact throughout the steps of the policy transaction.

In submission and issuance policy transactions, contacts are always synchronized with the account. In cancellation policy transactions, contacts are never synchronized with the account. Other policy transactions synchronize to the account if both of the following are true:

- The effective date of the policy transaction is not in the future
- At the beginning of the policy transaction, the last update time of the contact is earlier than or equal to the effective date of the policy transaction

If a policy is synchronized with the account, a change to contact and contact role information outside the policy impacts the following:

- **Pending Policy Transactions** – Changes are immediately apparent when viewing any pending policy transactions because those policy transactions always display the up-to-date information. This information comes from the associated account-level contact and role information.
- **Quoted Policy Transactions** – When you bind a quoted but not bound submission, the application verifies that the revised policy contact information matches the account contact information. The information will not match if someone made a change to the synchronized information on the account contact after the quote was made. An example of this is contact name. If the contacts do not match, you will see a validation error. PolicyCenter generates this validation error because the change could affect the forms generated at quote time. When you quote the policy again, the application synchronizes the contact.

- **New Policy Transactions** – When new policy transactions on existing policies begin, the contacts on those policy transactions always reference the most recent contact and role information. It does not matter what contact and role the revision they are based on references.

Contact Revisioning in Future-Dated Changes

If the effective date of the policy transaction is in the future, changes to revised fields in contacts are handled as future-dated changes.

Note: This revisioning behavior does not apply to submission or issuance policy transactions. These policy transactions are always synchronized with the account.

At the beginning of a future-dated change, PolicyCenter copies down contact information from the account, so that the policy transaction starts out with the most up-to-date information. During the policy transaction, the account and policy contacts are not synchronized. They are not synchronized because changes to the contact are in the future and the account data must represent the current contact information. During the policy transaction, changes to the policy contact data create work items for the **Apply Pending Account Data Updates** batch process. In the default configuration, this batch process runs nightly. On the day that the change takes effect, the batch process updates the account contact. For more information, see “Contact Batch Process” on page 407.

Example of a Future-Dated Change

On September 1, 2010, Jane Smith calls to say she is getting married on September 4, 2010 and will change her name to Jane Smith-Jones. The agent issues a future-dated policy change updating her marital status and name. PolicyCenter rates the policy again because of the marital status change.

Until September 4, the account contact remains Jane Smith, a single woman. On September 4, the **Apply Pending Account Data Updates** batch process updates the account contact to Jane Smith-Jones, a married woman.

Contact Revisioning in Back-Dated Changes

If the effective date of the policy transaction is earlier than the last update time of the contact, changes to revised fields in contacts are handled as back-dated changes.

Note: This revisioning behavior does not apply to submission or issuance policy transactions which are always synchronized with the account.

When starting a back-dated policy change, PolicyCenter does not copy contact information from the account because this information is newer. That is, the last update time of the contact is more recent than the effective date of the policy change. So for earlier points in time, the based-on policy contact data is more accurate. During the policy transaction, the account and policy contacts are not synchronized. They are not synchronized because that might result in changing data at the account level that is more current than in the back dated change.

When contact information changes in a back-dated change, PolicyCenter creates an activity and note reminding the user about the change. The user can choose to apply the change manually. PolicyCenter does not apply the change automatically because PolicyCenter cannot assume that the change will always be applied.

Example of a Back-Dated Change

This example makes further changes to Jane Smith-Jones’ policy. Previously, the agent made future-date changes to the policy in “Example of a Future-Dated Change” on page 373.

On September 10, 2010, Jane calls to report that since the beginning of the policy, her date of birth has been incorrectly entered as May 1, 1990. She was actually born in 1988. The agent issues a policy change that corrects the date of birth effective back to the beginning of her policy on July 1, 2010. This change is an out-of-sequence policy change. On the policy period from July 1 through September 3, PolicyCenter updates Jane Smith’s date of birth to 1988. On the policy period from September 4 forward, PolicyCenter updates Jane Smith-Jones’ date of birth to 1988.

PolicyCenter does not update the account contact automatically in a back-dated change. PolicyCenter creates an activity and note describing the change to the contact. The user assigned to the activity decides whether to make that change, and updates the account contact directly. In the case of Jane's date of birth, the user gets the activity. The user decides to update date of birth on the account contact because it is true throughout the life of the policy.

In some back-dated changes, the user does not want to update the account contact. Suppose on September 10, 2010, Jane calls to report that since the beginning of the policy her marital status has been incorrectly recorded as Single. She was married previously, and her correct status is Separated. In the policy, the agent issues a policy change that corrects the marital status effective back to the beginning of her policy on July 1, 2010. This change is an out-of-sequence policy change. On the policy period from July 1 through September 3, the last update date on the marital status field is July 1. Therefore, PolicyCenter updates Jane Smith's marital status to Separated on this policy period. On the policy period from September 4 forward, the last update date on the marital status is later than the effective date of the back-date change. Therefore, PolicyCenter does not change Jane Smith-Jones' marital status of Married on this policy period. PolicyCenter creates an activity and note describing the change of marital status. In the case of Jane's marital status, the user gets the activity and decides not to update the marital status on the account contact.

Note: A back-dated policy change is not necessarily an out-of-sequence policy change. For example, assume there is a policy which has no policy changes on it. You can make a series of back-dated policy changes moving forward in time. These policy changes are not out-of-sequence.

Linking an Address Between Multiple Contacts

Two or more contact often have the same address. For example, a personal auto policy has four drivers: the husband and wife and their two teenage children. Each driver on the policy has separate contact information with an address. Because all drivers live at the same address, the address information is the same. In PolicyCenter, to avoid retyping the address, you can link a group of addresses that are the same.

If you change the address information for one contact, you can:

- Update the address for all contacts in the linked group.
- Update the address for this contact only, and remove it from the linked group. PolicyCenter removes the linked group if only one contact remains in the group.

You can update linked addresses from an external system. The API provides methods for updating the linked address on a contact.

In the default configuration, you can link to addresses on the following types of contacts:

- Primary named insured
- Account holder
- Named insured

You can configure PolicyCenter to link to other types of contacts.

See also

- “Working with Linked Addresses” on page 385
- “Linked Addresses Object Model” on page 391
- “Configuring Linked Addresses for Contacts” on page 405
- “PolicyCenter Linked Addresses and Side Effects” on page 32 in the *Contact Management Guide*

Working with the Contact Tab

The following topics describe the **Contact** tab functionality and implementation details:

- “Creating a New Contact on the Contact Tab” on page 375

- “Searching for a Contact on the Contact Tab” on page 375
- “Selecting Recently Viewed Contacts on the Contact Tab” on page 376
- “Viewing Contact File Details on the Contact Tab” on page 377
- “Creating a New Account from a Contact on the Contact Tab” on page 377
- “Viewing Accounts Associated with a Contact on the Contact Tab” on page 377
- “Viewing Policies Associated with a Contact on the Contact Tab” on page 377
- “Viewing Policy Transactions Associated with a Contact on the Contact Tab” on page 378
- “Viewing Claims Associated with a Contact on the Contact Tab” on page 379
- “Viewing Billing Information for a Contact on the Contact Tab” on page 379
- “Viewing the Account Holder Info Screen” on page 380
- “Contact Tab Behavior” on page 382

See also

- “Centralized View of Contacts on the Contact Tab” on page 370
- “Configuring the Contact Tab” on page 408

Creating a New Contact on the Contact Tab

1. Select **Contact** → **New Contact** and choose either **New Company** or **New Person**. The **New Contact** screen displays different fields depending upon whether the contact is a person or a company.

2. Add the primary address on the **Contact Detail** tab. On the **Addresses** tab, enter additional addresses, set the primary address, or remove addresses.

The **Check for Duplicates** button appears if the following are true:

- You are adding a new contact.
- You are connected through a plugin that supports checking for duplicates. In the default configuration, the `DemoContactSystemPlugin` and `ABContactSystemPlugin` support this.
- The current contact is not linked to a contact in the contact management system. A contact is linked to a contact in the contact management system if both contacts have the same `AddressBookUID`.

3. If **Check for Duplicates** appears, click this button to verify that the contact does not already exist in the contact management system.

If PolicyCenter finds duplicate contacts, you can **Select** one. The selected contact replaces the new contact. Any contact information for the new contact is overwritten. Alternately, you may decide that this is not a duplicate, and click **Return to New Contact**.

4. Click **Update** or **Cancel**.

If you did not click **Check for Duplicates**, PolicyCenter checks for duplicates when you click **Update** to create the new contact. PolicyCenter displays the duplicate contacts.

If you click **Update**, PolicyCenter displays the **Contact File Details** screen for the new contact.

If you click **Cancel**, PolicyCenter discards the changes and displays the **Search Contacts** screen.

See also

- “Detecting Duplicates When Integrated with ContactManager” on page 725

Searching for a Contact on the Contact Tab

You can search for a contact by using the **Contact** → **Search** menu item. This menu item and the **Search** → **Contacts** menu item display the **Search Contacts** screen.

The **Search Contacts** screen displays a list of matching contacts. Choose **Company** or **Person** from the **Type** drop-down list. If you select **Company**, PolicyCenter displays a **Name** field. If you select **Person**, PolicyCenter displays a **First name** and **Last name** field.

For a company, you must specify one of the following:

- **Name**
- **Tax ID (EIN)**
- **ZIP Code**
- **City and State**

For a person, you must specify one of the following:

- **First name**
- **Last name**
- **Tax ID (SSN)**
- **ZIP Code**
- **City and State**

If you are integrated with an external contact management system, this search returns both contacts in PolicyCenter and contacts that are only in the external contact management system.

If you are integrated with ContactManager, ContactManager returns contacts with the **Client** tag. The code that specifies that ContactManager only returns contacts with the **Client** tag is in the **ABContactSearchCriteriaInfoEnhancement** class. Modify this class if you want ContactManager to return contacts with other tags.

If the contact is in PolicyCenter, PolicyCenter displays complete information about the contact, such as accounts, policies, and policy transactions associated with the contact. If the contact is only in the contact management system, you can do one of the following:

- View the contact details. You cannot edit the contact.
- Create a new account with that contact as the account holder. As a result, the contact is added to PolicyCenter.

See also

- “Searching for Contacts” on page 77 in the *Contact Management Guide*

Selecting Recently Viewed Contacts on the Contact Tab

Recently viewed contacts appear at the bottom of the **Contact** drop-down menu. Select one of these contacts to displays the **Contact File Details** for that contact.

For each user, the recently viewed list is initially empty. Contacts are added as the user views contact details over multiple sessions.

If you select a recently viewed contact, PolicyCenter displays the details for that contact, and moves that contact to the top of the list. If, the contact no longer exists, PolicyCenter displays the **Search Contacts** search screen, and removes the contact from the list. For example, a recently viewed contact no longer exists if that contact was merged into another contact.

Contacts that exist only in external systems are not added to the list of recently viewed contacts. However, if you create an account for an external contact, that contact is added to PolicyCenter and appears on the list of recently viewed contacts.

More recently viewed contacts appear higher on the list. When the maximum number of recent contacts has been reached, older contacts are removed and replaced by newer ones.

Each user can specify the maximum number of recent contacts in **Preferences**. For more information, see “Setting Preferences” on page 40.

Viewing Contact File Details on the Contact Tab

If you are viewing a contact, you can click the **Details** link in the left sidebar to view the **Contact File Details** screen. This screen displays basic information such as name, addresses associated with the contact, and official IDs. Click **Edit Contact** to make changes.

If PolicyCenter is integrated with a contact management system, this screen displays information retrieved from the contact management system.

If multicurrency display is enabled, the **Contact File Details** screen has a **Preferred Currency** field in the **Address** section. This field displays the preferred settlement currency on the contact (`Contact.PreferredSettlementCurrency`).

When you create a new contact, the **Preferred Currency** updates when you change the **Country** field on **Address**.

Creating a New Account from a Contact on the Contact Tab

If you are viewing a contact, select **Actions** → **New Account** to create an account with the current contact as the account holder. If you select this menu item, PolicyCenter displays the **Create account** screen. PolicyCenter populates most of the fields on the **Create Account** screen with values from the contact. You must select a producer.

Note: This screen is the same screen that PolicyCenter displays when you select **Account** → **New Account** then select **Create New Account**.

Fill in the information under **Select Producer** and click **Update**. PolicyCenter creates a new account with the original contact as the **Account Holder**. If the contact was an external contact, PolicyCenter creates its own copy of this contact.

If you click **Cancel**, PolicyCenter displays the **Enter Account Information** screen.

Viewing Accounts Associated with a Contact on the Contact Tab

If you are viewing a contact, click **Accounts** in the left sidebar to view the **Contact File Accounts** screen. This screen displays summary information for each account on which the contact appears on the **Account File Contacts** screen. The summary information includes the account number, first and last name or company name, primary address, primary phone, and email address. The summary information matches the contact's information if the contact is the account holder for the account. The **Roles** column displays the roles the contact holds on the account.

Note: This screen displays the accounts for which you have sufficient producer code permissions.

If you have sufficient permissions to view the account, the **Account #** is a link to the account file.

If an account appears in the list, that Account has an **AccountContact** that points to the current **Contact**.

Viewing Policies Associated with a Contact on the Contact Tab

If you are viewing a contact, click **Policies** in the left sidebar to view the **Contact File Policies** screen. This screen displays a list of policies on which the contact plays a role.

Note: This screen displays the accounts and policies for which you have sufficient producer code permissions.

You can click the **Account #** or **Policy #** link to view the corresponding account or policy file. If the policy is archived, PolicyCenter displays the archived policy page. PolicyCenter does not check the permissions for the account and policy because it checks permissions when creating the list.

The list view displays the account number of the policy, the policy number, the policy type, status, the start date for the policy, and the effective and expiration dates.

The list can be filtered by the status of each period and product.

The **Status** drop-down list contains the following choices:

- **In Force/Scheduled** – Default. Display in-force policies and bound policies that have not yet started.
- **Expired**
- **Canceled**
- **All**

The **Product** drop-down list contains the following choices:

- **All** – Default
- Any products in the unfiltered results. For example, the contact has personal auto and business auto policies. Those two products appear in the drop-down list, but inland marine does not.

If a policy appears on this screen, that **PolicyPeriod** has a **PolicyContactRole** that points to the current **Contact** through the **ContactDenorm** field.

Viewing Policy Transactions Associated with a Contact on the Contact Tab

If you are viewing a contact, click **Policy Transactions** to view the **Contact File Policy Transactions** screen. This screen displays a list of policies on which the contact appears on the **Contacts** screen in the policy transaction. Click the **Transaction #** link for detailed information.

Note: This screen displays the policies and policy transactions for which you have sufficient producer code permissions.

The **Policy #** column is a link to the policy file for all policy transaction types that are based on an existing policy period. The only excluded policy transaction types are submission and rewrite new account. If the policy is archived, PolicyCenter displays the archived policy page.

Click the **Transaction #** to jump to the corresponding policy transaction wizard. If the policy is archived, PolicyCenter displays the archived policy page. Permissions for the policy and policy transaction are not checked because permissions control the policies and policy transactions that the list displays.

You can filter the list view by status, policy transaction type, and product.

The **Status** drop-down list contains the following choices:

- **All** – Default
- **Open**
- **Complete**

The **Type** drop-down list lets you select a type of policy transaction:

- **All** – Default
- **Submission**
- **Cancellation**
- **Renewal**
- **Policy Change**
- **Reinstatement**
- **Rewrite**
- **Rewrite New Account**
- **Audit**

The **Product** drop-down list contains the following choices:

- **All** – Default
- Any products in the unfiltered list of results

If a policy transaction appears on this screen, then the **PolicyPeriod** has a **PolicyContactRole** that points to the current **Contact** through the **ContactDenorm** field.

Viewing Claims Associated with a Contact on the Contact Tab

If you are viewing a contact, click **Claims** to view the **Contact File Claims** screen. This screen displays claim information retrieved from a claim system. You can configure this screen to display information retrieved from a third-party claim system. If the integration with Guidewire ClaimCenter is enabled, this screen displays claims from ClaimCenter.

To view this screen, you must have the **View contact file claims** permission. The code for this permission is `contactclaims`.

Enter search criteria on this screen to search an external claim system for claims on all policy periods related to the current contact. PolicyCenter displays claims filed against policies on which the contact has a role.

PolicyCenter displays a claim if you have sufficient permissions and producer code security.

The **Claim Number** column is a link to the claim system if:

- The `ClaimSystemURL` parameter is specified in `config.xml`.
- The claim is not archived.
- The current user has the **View claim system** permission. The code for this permission is `viewclaimsystem`.
- For restricted claims, the current user must also have the **View restricted claim** permission. The code for this permission is `viewrestrictedclaim`.

The **Policy Number**, **Product**, and **Insured** columns come from the policy period associated with the PolicyCenter `Claim` object. The claim search plugin returns the `Claim` object. The remaining columns come from the `Claim` object and correspond to the `LossDate`, `ClaimNumber`, `Status`, and `TotalIncurred` fields.

This screen is identical to the **Account File Claims** screen with the following differences:

- The list of policy numbers to search is based off the contact rather than the account.
- The search results includes an **Insured** column but does not have a **Policy Period** column.
- The **Claim Details** tab for the selected claim uses the same PCF file as **Account File Claims** screen but hides the **Policy Number** and **Product** fields.

If a policy appears on this screen, that `PolicyPeriod` has a `PolicyContactRole` that points to the `Contact` through the `ContactDenorm` field.

See also

- “Claim System Integration” on page 715
- “Claim Search from PolicyCenter” on page 557 in the *Integration Guide*
- “Enabling Integration between ClaimCenter and PolicyCenter” on page 95 in the *Installation Guide*

Viewing Billing Information for a Contact on the Contact Tab

If you are viewing a contact, click **Billing** in the left sidebar to view the **Contact File Billing** screen. This screen displays billing information for policies retrieved from a billing system. You can configure this screen to display information retrieved from a third-party billing system. If the integration with Guidewire BillingCenter is enabled, this screen displays billing information from BillingCenter.

PolicyCenter finds all policies on which the contact is an **Account Holder** or **Billing Contact** and for which the current user has view permissions. PolicyCenter retrieves information about the account and subaccounts from an external billing system. The information includes the subaccount numbers for every account associated with the contact, and information about each subaccount.

The first three columns are:

- **Policy Account #** – The PolicyCenter account. This column is a link to the corresponding account file in PolicyCenter.
- **Policy Role** – Lists the roles that the contact holds on that account.

- **Billing Account #** – The billing account or billing subaccount number. Depending upon the integration, this may be the same as the **Policy Account #**.

If the BillingCenter integration is enabled, this is the billing account or subaccount number from BillingCenter. If this is the main billing account, the account number is the same as **Policy Account #**.

This column is a link if:

- The `BillingSystemURL` parameter is specified in `config.xml`.
- The user has the `View billing system` permission. The code for this permission is `viewbillingsystem`.

If the BillingCenter integration is enabled, the link opens the account in BillingCenter.

The remaining columns come from the billing account information (`BillingAccountInfo` class) returned by the billing system. These columns are:

- **Account Name** – The name on the account. The `AccountName` property on `BillingAccountInfo`.
- **Status** – Displays **Delinquent** in red or **In Good Standing**. The `Delinquent` property.
- **Billed Outstanding** – The billed outstanding amount. The `BilledOutstandingTotal` property.
- **Current Due** – The current amount due. The `BilledOutstandingCurrent` property.
- **Past Due** – The amount past due. The `BilledOutstandingPastDue` property.
- **Unbilled** – The amount not billed. The `UnbilledTotal` property.

See also

- “Billing System Integration” on page 701
- “Billing Integration” on page 509 in the *Integration Guide*
- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*

Viewing the Account Holder Info Screen

The **Account Holder Info** screen provides a summary of a PolicyCenter account holder, including information retrieved from BillingCenter, ClaimCenter, and ContactManager, or the equivalent third-party systems of record. This screen provides a snapshot of the account holder’s value to the organization.

When integrated with BillingCenter, ClaimCenter, and ContactManager (or equivalent third-party systems of record), the screen displays data across all accounts for which the contact is the account holder. This information includes:

Field	Description
Account Holder Info	<ul style="list-style-type: none"> • Account Holder – The name of the person or company • Accounts – Number of accounts on which the contact is the account holder, with link to a summary listing all accounts for the account holder. • In-Force Premium – Amount across all policies on those accounts. <p>The following fields provide a billing summary for the accounts.</p> <ul style="list-style-type: none"> • Total Unbilled – As an amount. • Total Currently Billed – As an amount. • Total past Due – As an amount. • Total Outstanding – As an amount.
Customer Value Metrics	<ul style="list-style-type: none"> • First Policy Effective Year – Calculated across all policies on the accounts. • Active Policies – Number across the accounts. • Canceled by Customer – Number of cancellations initiated by the customer in the past 12 months. • Canceled by Carrier for Non-Payment – Number of cancellations initiated by the carrier for non-payment in the past 12 months. • Other Cancellations – Number of other cancellations initiated in the past 12 months. • Lifetime Premium – As an amount. • Open Claims Count – Number of open claims. • Net Total Incurred – Open claims net total incurred.
Pending Policy Transactions	<ul style="list-style-type: none"> • Create Date – The creation date of the transaction. • Policy # – The policy number of the transaction. • Product – The product associated with the transaction. • Transaction # – With a link to the transaction. • Type – The type of the transaction. • Status – The status of the transaction.
Open Claims	<ul style="list-style-type: none"> • Policy Number – A link to the Policy Summary screen for the policy associated with this claim. • Product – The product associated with the policy. • Insured – The name of the insured. This may be different than the account holder. • Loss Date – The loss date. • Claim Number – The claim number. • Status – The status of the claim. • Total Incurred – The total incurred for this claim.

The **Account Holder Info** link appears in the left sidebar of the **Contact** tab only when the selected contact has the **Account Holder** contact role on at least one account. PolicyCenter uses the set of accounts on which the contact is the account holder to retrieve associated policy, claim, and billing information.

You must have the **Account Holder Info** (`viewaccountholderinfo`) permission. enables administrators to implement role-based access to the new screen. Even if you have permission to view the **Account Holder Info** screen, you also must have individual permissions to view the data elements the screen presents. Otherwise, the screen omits that data.

You can modify the default configuration and add screens that display information for other account contact roles such as a **Secondary Account Holder** role.

See also

- “Account Holder Information” on page 371
- “Configuring the Account Holder Info Screen” on page 477 in the *Configuration Guide*

To view the Account Holder Info screen

1. If not already running, start the integrated billing, contact management, and claim processing systems.
2. Start PolicyCenter.
3. Log into PolicyCenter as su and grant the **Account Holder Info** permission to appropriate user roles. The code for this permission is `viewaccountholderinfo`.
4. Log into PolicyCenter as a user who has the `viewaccountholderinfo` permission.
5. In the **Contact** tab, search for a contact that you know is an account holder. If you are using the sample data set, you can search for and select “Ray Newton”. The **Contact** tab shows an **Account Holder Info** link in the left sidebar.
6. Click the **Account Holder Info** link to display the **Account Holder Info** screen.
7. Verify that the **Account Holder Info** link does not appear when the selected contact is not an account holder. If you are using the sample data set, you can search for and select “Stan Newton”.

Contact Tab Behavior

If you click the **Contact** tab, PolicyCenter displays one of the following:

- The **Contact Search** screen if you have not viewed any contacts
- The **Contact File Details** for the most recently viewed contact

This behavior is defined in the `action` property in the PCF file that defines the **Contact** tab.

Working with Contacts in Policies and Accounts

In the user interface, you can view all contacts and contact roles that are related to an account. This topic describes, from the user’s perspective, how to work with contact information in a policy or account.

- “Adding a Contact to an Account” on page 382
- “Adding a Contact From the Address Book” on page 383
- “Editing a Contact” on page 384
- “Removing a Contact from an Account” on page 384
- “Changing the Active Status on a Contact from an Account” on page 384
- “Adding or Removing a Contact from a Policy” on page 385
- “Adding or Removing a Contact Role” on page 385
- “Working with Linked Addresses” on page 385
- “Changing Revised Contact Information in a Policy Change” on page 386

Adding a Contact to an Account

You can add a contact to an account. The newly created contact is available for new and existing policy transactions. This topic provides step-by-step instructions for adding a contact to an account.

You can also add a contact:

- While working on a policy transaction.
- From the **Contact** tab on the tab bar. For instructions, see “Creating a New Contact on the Contact Tab” on page 375.

To add a contact to an account

1. Navigate to an account.

2. Click **Contacts** in the left sidebar. The **Account File Contacts** screen appears.
3. Click **Create New Contact**. Select the type of contact and whether the contact is:
 - **New Company**
 - **New Person**
 - **From Address Book**
4. If you selected:
 - **New Company** or **New Person**, continue to the next step
 - **From Address Book**, see “[Adding a Contact From the Address Book](#)” on page 383
5. Fill in the necessary information. At this point, you can also add additional roles to the contact.
6. Click **Update** to save.

Adding a Contact From the Address Book

You can add a contact from the address book to accounts and policies. When you search the address book, you can select contacts from the PolicyCenter internal address book. If you are integrated with ContactManager or another contact management system, the search also includes contacts from that system.

When you select **From Address Book** in PolicyCenter, you can select contacts from the internal address book and from the contact management system.

If the contact is currently in the contact management system, then PolicyCenter pulls the contact into its internal address book when you **Select** it.

In PolicyCenter, you can search for contacts from a contact management system in the following places:

- **Contact** tab → **Search** menu item.
- **Search** tab → **Contact** menu item.
- **Account File Contacts** screen – You can choose to create a new contact from the address book.
- **Policy job** screens – You can choose to create a new contact from the address book.

To add a contact from the address book

1. Add a contact to an account or policy.

- To add a contact to an account, navigate to the **Account File Contacts** screen. Select **Create New Contact** → **Type** → **From Address Book**.
- To add a contact to a policy, you can access the **From Address Book** list item in various places such as the **Drivers** screen in a personal auto policy. Select **Add** → **From Address Book** to add an additional driver.

The **Search Address Book** screen appears when you select the **From Address Book** list item. The **Search Results** displays a list of contacts.

2. From the **Type** drop-down list, select **Company** or **Person**.

If PolicyCenter is integrated with ContactManager, the ContactManager search is limited to a contact type of **Person** or **Company** with the **Client** tag. In the ContactManager object model, these are the **ABPerson** and **ABCompany** subtypes of **ABCContact**.

3. Enter other search criteria and click **Search**.

In **Search Results**, PolicyCenter displays the **Name**, **Address**, **Phone**, and **Email** for each contact. If the **External** column is **Yes**, the contact exists only in the external contact management system. If the **External** column is **No**, the contact is internal to PolicyCenter and may also exist in the external contact management system.

4. Click **Select** in the first column of the contact in the **Search Results** list.

If the contact is external only, the contact is retrieved from the contact management system, and an internal PolicyCenter contact is created.

When you add an account contact, the new contact is added to the **Account File Contacts** page.

When you add a policy contact, an account contact is also created.

If you **Update** the contact, PolicyCenter pushes your changes to the contact management system.

Editing a Contact

You can modify contacts on policies, accounts, and on the **Contact** tab. When you modify a contact on the **Contact** tab or in the context of an account, PolicyCenter saves the modified contact information immediately.

PolicyCenter also save the modified contact information across all accounts which use the contact. When you modify a contact in the context of a policy transaction, the modified contact information is always saved on the policy, and sometimes flows up to the account level. For more information, see “[Revisioning Contact Information in Policies](#)” on page 371.

To modify contact information on an account:

1. On the **Account File Contacts** screen, select the chosen contact link.
2. Make necessary edits on the **Edit** screen. This screen has the following tabs:
 - **Contact Detail** – Update name, address and official IDs
 - **Roles** – Add roles to the contact
 - **Addresses** – Specify primary address, provide additional addresses
3. Click **Update**.

Removing a Contact from an Account

Note: Removing a contact from a policy removes it from the policy, but not the account. Removing a contact from an account makes it unavailable for future policy transactions.

Follow these steps to remove a contact from an account:

1. On the **Account File Contacts** screen, select the check box of the contact that you want to remove.
2. Click **Remove Contact**.

PolicyCenter prevents you from removing the account holder and contacts that are associated with any policy or policy transaction.

Changing the Active Status on a Contact from an Account

You can change the active status on a contact in an account. An inactive contact can no longer be used in new or renewed policies. An inactive contact does appear in the list of contacts that can be added to new policy terms. PolicyCenter removes an inactive contact from new policy transactions based on existing policy terms.

To change the active status on a contact from an account

1. On the **Account File Contacts** screen, select the check box of the contact on which you want to change the active status.
2. Click **Change Active Status**.

Note: You cannot change the active status on the account holder.

Adding or Removing a Contact from a Policy

If there are contacts on the account, you can add or remove them from a policy during a policy transaction such as a submission or change policy transaction. For example, on the **Policy Info** screen, under the **Named Insureds** section, you can:

- Create a new named insured. This action also adds the contact to the account.
- Add or remove an existing contact as a named insured on the policy

Adding or Removing a Contact Role

1. On the **Account File Contacts** screen, select the contact that you want to modify.
2. On the **Contact Detail** card, under **Roles**, click **Add Role** and select the type of role.
3. To remove a role, select the role you want to delete and click **Remove Role**.

You can only remove roles if the contact does not play that role on any bound policy or policy transaction.

Working with Linked Addresses

In PolicyCenter, if a contact has the same address as another, you can link the addresses to avoid retying the address. A change to one address updates the other linked addresses in the group. You can link an address on a new or existing contact. In the default configuration, the contact must be a primary named insured, named insured, or account holder.

To link to an address on another contact

1. In PolicyCenter, navigate to a contact or create a new contact.
2. In the **Address** section on the **Contact Detail** tab, select the **Same address as** drop-down list.
 - The drop-down list displays the account holder and named insureds on the account, and their addresses. The list does not include the current contact.
 - An asterisk appears before an address that is the primary address.

Note: The **Same address as** drop-down list does not appear if there is no contact to display. For example, the list does not appear if you are editing the account holder and the only potential linkable contact is the account holder.

3. Select an address.
 - The address is copied into the address fields.
 - The address section displays **This address is linked to other addresses**.
 - The address is no longer editable.
4. Click **Update** to save your changes.

Note: You can also link to an address on the **Addresses** tab of the contact.

To edit a linked address

A linked address displays **This address is linked to other addresses** above the **Address** label.

1. On the **Change to** drop-down list, select **Edit address**.

PolicyCenter displays the **Address Detail** screen. At the top of this screen, the **Contacts Using this Address** section displays the linked contacts.

Note: If you did not click **Update** to save your changes when you linked the address, then those contacts may not appear on the **Contacts Using this Address** section.

2. Make changes to the address.

3. Save your changes by clicking one of the following buttons:

- **Update All Linked Addresses** – PolicyCenter updates the linked addresses with your changes.
- **Update Only This Address and Unlink** – PolicyCenter unlinks the address from the group, and saves your changes to the address for this contact only. If the linked address group contains only one address, then PolicyCenter removes the linked group.

See also

- “Linking an Address Between Multiple Contacts” on page 374
- “Linked Addresses Object Model” on page 391
- “Configuring Linked Addresses for Contacts” on page 405

Changing Revised Contact Information in a Policy Change

This topic provides step-by-step instructions on making changes to revised contact information. The example uses a personal auto policy change.

- “Changing Revised Contact Information in a Future-Dated Policy Change” on page 386
- “Changing Revised Contact Information in a Back-Dated Policy Change” on page 388
- “Viewing Revised Contact Information in the Account” on page 389

Changing Revised Contact Information in a Future-Dated Policy Change

These instructions show how to change revised contact information in a future-dated policy change. The example modifies the driver’s last name on a personal auto policy. This example uses the Ray Newton account and personal auto policy in the large sample data. Log in as the user aapplegate unless otherwise specified.

To modify the account contact

1. In PolicyCenter, navigate to the Ray Newton account.
2. Click **Contacts** in the left sidebar and click **Ray Newton** in the list of contacts.
The contact’s first name is a revised field.
3. Change Ray’s First Name to Raymond and click **Update**.
4. Navigate to the personal auto policy on this account.

Notice that the **Account Name** is **Raymond Newton**, but the **Primary Named Insured** on the bound policy is **Ray Newton**.

To create a future-dated change to revised contact information

1. In PolicyCenter, navigate to the Ray Newton personal auto policy. This policy is in force, and Ray’s first name has not been updated to Raymond.
2. Select **Actions → Change Policy**.
3. On the **Start Policy Change** screen, set the **Effective Date** to a day in the future. The day must be within the current policy period. For example, set the **Effective Date** one week in the future.
4. In **Description**, enter “Future-dated change to contact’s name and marital status”.
5. Advance to the **Policy Info** screen.

Notice that the name of the **Primary Named Insured** has been changed to **Raymond Newton** because the contact information was copied down at the start of the policy transaction.

6. Advance to the **Drivers** screen.

In the next step, you will make the future-dated change to revised contact information.

7. Change the **Marital Status**. In the Ray Newton policy, there is one driver named John Smith. Change John Smith's **Marital Status** to **Married**.
8. Change the name of the driver. In the Ray Newton policy, change John's **Last Name** to **Smith-Jones**.
9. Click **Quote**.
10. Click **Issue Policy**.

To view a future-dated change to revised contact information

1. On the **Policy Change Bound** screen, click **View your policy** or navigate to the policy.
2. In the **Date** field on left sidebar, enter the effective date of the policy change and press **ENTER**.
3. In the left sidebar, click **Drivers**. The last name of the driver is updated. If you changed the Ray Newton policy as described in the previous steps, the driver's name is John Smith-Jones and he is married.
4. In the **Date** field on left sidebar, enter today's date and press **ENTER**.
5. In the left sidebar, click **Drivers**. The last name of the driver is not updated. If you changed the Ray Newton policy, the driver's name is John Smith and he is single.

To enable the internal debug tools

Running the batch process in the next series of steps requires that you use the **Run Clock** command. This command is available if internal debug tools are enabled in PolicyCenter.

1. In Studio, edit **Other Resources → config.xml**.
2. Set the value of the **EnableInternalDebugTools** parameter to **true**.
3. Save your changes.
4. Restart PolicyCenter.

To update the account contact with future-dated change to revised policy contact information

These steps require that you have enabled the internal debug tools.

1. Log in as “su”.
Note: The su privileges are necessary for using the run command and for running the batch process.
2. Navigate to the Raymond Newton account. This account is the account of the policy with changes to revised contact information.
3. In the left sidebar, click **Contacts**.
4. In **Account File Contacts**, click **John Smith**, the contact that you changed.
Notice that the future-dated change to name and marital status are not updated.
5. In the **QuickJump** box, enter “**Run Clock addDays N**” where **N** is the number of days that the future-dated policy change goes into effect. Enter “**Run Clock addDays 7**”.
6. Press **ALT+SHIFT+T** to display **Server Tools**.
7. On the **Batch Process Info** screen, click **Run** in the **Apply Pending Account Data Updates** batch process.
In the default configuration, this batch process runs nightly.
8. Select **Actions → Return to PolicyCenter**.
9. Navigate to the Raymond Newton account.
10. In the left sidebar, click **Contacts**.

In **Account File Contacts**, notice that the account contact name has been updated. John Smith is now John Smith-Jones.

11. Click the contact that you changed.

Notice that the future-dated change to name and marital status are updated.

12. Click **Log Out** to log out as su.

Changing Revised Contact Information in a Back-Dated Policy Change

These instructions show how to change revised contact information in a back-dated policy change. The example modifies the driver's date of birth on a personal auto policy.

To create a back-dated change to revised contact information

- 1.** In PolicyCenter, log in as "aapplegate".
- 2.** Navigate to the Raymond Newton personal auto policy. This policy is in force. Note the **Period Eff Date** of the submission.
- 3.** Select **Actions → Change Policy**.
- 4.** On the **Start Policy Change** screen, set the **Effective Date** to the beginning of the policy period. This date is the **Period Eff Date** you noted in the previous step. The day must be earlier than the current date.
- 5.** In **Description**, enter "Back-dated change to contact's date of birth".
- 6.** Click **Next**.
PolicyCenter displays a popup message warning that this is an out-of-sequence transaction.
Click **OK**.
- 7.** Advance to the **Policy Info** screen. Notice that the **Primary Named Insured** is Ray Newton.
- 8.** Advance to the **Drivers** screen.
Notice that the driver is John Smith and single man. In the next step, you will make the back-dated change to revised contact information.
- 9.** Change the **Date of Birth**. In the Ray Newton policy there is one driver named John Smith born on 01/01/1970. Change his **Date of Birth** to "01/01/1967".
- 10.** Click **Quote**.
- 11.** Click **Issue Policy**.

To view a back-dated change to revised contact information

- 1.** On the **Policy Change Bound** screen, click **View your policy** or navigate to the policy.
- 2.** In the **Date** field on left sidebar, enter the effective date of the policy change and press **ENTER**. The effective date of the policy change was the start of the policy period.
- 3.** In the left sidebar, click **Drivers**. The driver's **Date of Birth** is updated. If you changed the Ray Newton policy as described in the previous steps, the driver's **Date of Birth** is 01/01/1967.
- 4.** In the **Date** field on left sidebar, enter last day of the policy period and press **ENTER**. For example, if the expiration date is 10/04/2011, enter "10/03/2011".
- 5.** In the left sidebar, click **Drivers**. The driver's date of birth is updated.

To create a back-dated change to revised marital status

- 1.** Create another back-dated change effective as of the start of the policy.

PolicyCenter displays a message that your policy change is an out-of-sequence policy transaction. The message also says that there are future policy transactions.

2. In **Description**, enter “Back-dated change to contact’s marital status”.

3. Change John Smith’s Marital Status from Single to Separated.

PolicyCenter displays a message that there are out-of-sequence conflicts that you must resolve prior to quoting.

4. Advance to the **Policy Review** screen.

5. Go to the **Change Conflicts** tab.

In the current policy change, the separated marital status conflicts with the later status of married. You do not want to override the future marital status.

6. In **Override Future Conflict**, select **No** then click **Submit**.

7. Quote and issue the policy.

[Viewing Revised Contact Information in the Account](#)

PolicyCenter automatically applies future-dated changes on revised fields in the policy contact through the **Apply Pending Account Data Updates** batch process. PolicyCenter does not apply back-dated changes automatically. The user must update the account contact manually.

These instructions assume that you completed:

- “[Changing Revised Contact Information in a Future-Dated Policy Change](#)” on page 386
- “[Changing Revised Contact Information in a Back-Dated Policy Change](#)” on page 388

To update the account contact with back-dated change to revised policy contact information

1. Log in as “aapplegate”.

2. Navigate to the Raymond Newton account of the policy. This account is the account with changes to revised contact information.

3. In the left sidebar, click **Contacts**.

4. In **Account File Contacts**, click **John Smith-Jones**. This contact is the contact that you changed.

Because you ran the **Apply Pending Account Data Updates** batch job, the contact’s name is updated and his marital status is married. PolicyCenter does not update the account contact with back-dated changes. Therefore, the date of birth is not changed from 1970 to 1967. The marital status is not changed from married to separated. Do not make any changes to the contact at this time.

PolicyCenter created activities and notes to remind you that you may want to update the account contact manually.

5. Click **Summary** in the left sidebar on the account or policy.

Under **Current Activities**, PolicyCenter created activities for the back-dated changes. The subject of the activities is **Contact John Smith was changed**.

6. Click the **Subject** of one of the activities.

At the top of the screen, PolicyCenter displays the account file contact.

At the bottom of the screen, PolicyCenter displays the **Activity Detail**. The **Description** says that changes have only been applied to the policy. If you want the changes applied to the account level contact, you must do that manually. On the right side of the screen, the **Text** of the **New Note** displays the change details.

7. For the marital status change, you do not want to update the contact. Click **Edit**, and **Complete** the activity.

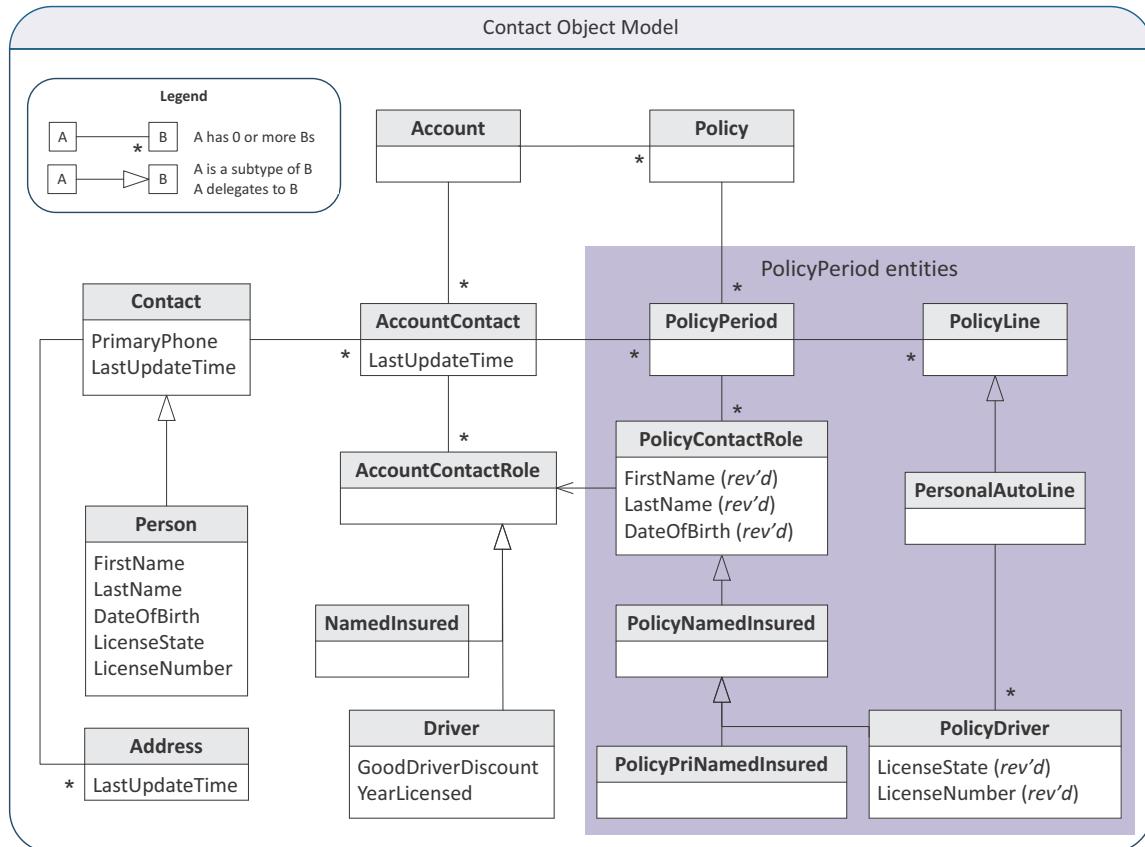
8. For the date of birth, make the change to the contact.

- Click **Edit**, and **Complete** the activity.
- At the top of the screen in **Contact Detail**, change John Smith's Date of Birth to "01/01/1967" and **Update** the contact.

Contact Object Model

PolicyCenter stores information about the contact on the account level and the policy level.

The contact object model is designed to handle contact revisioning. The following illustration shows some the basic relationships of the Contact entity, using personal auto as an example. Other lines of business have the same basic entity structure with their own PolicyLine subtype and fields.



The **Contact** entity has a number of subtypes including **Person** and **Company**. The diagram shows some of the fields for a **Person**. The **Contact** entity has **FirstName** and **LastName** fields from its subtype **Person** entity.

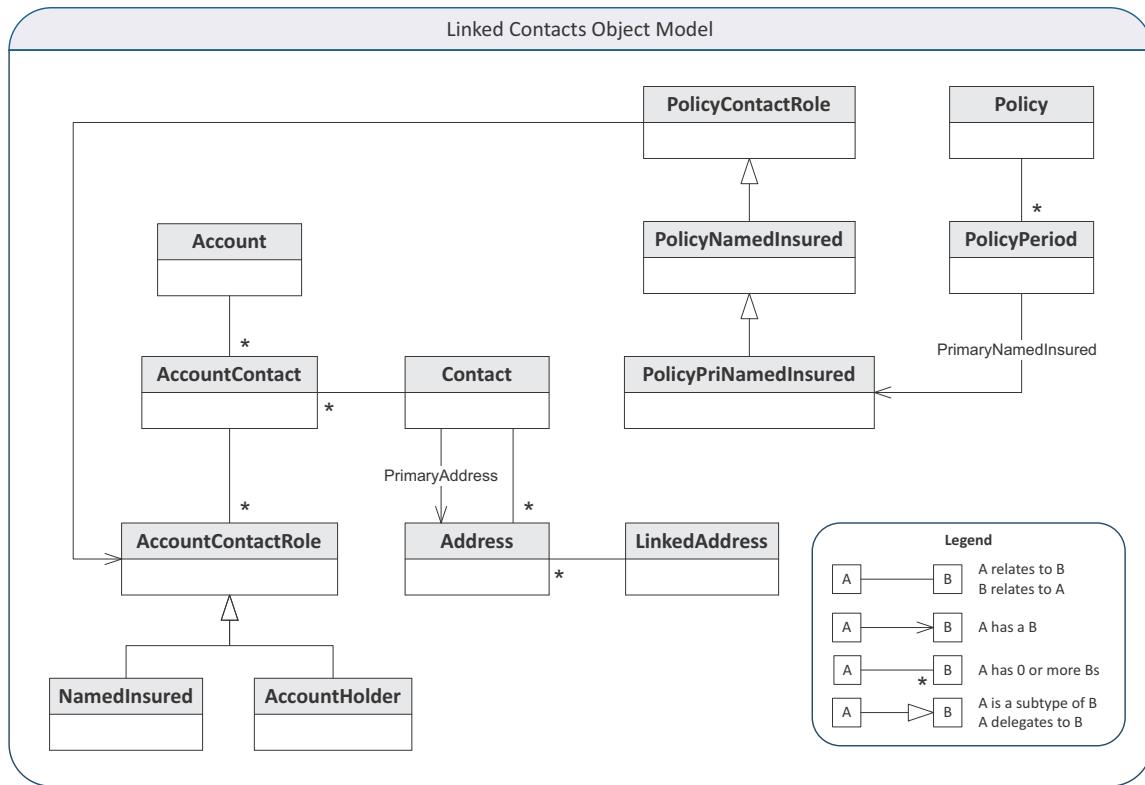
Some of the entities inside the **PolicyPeriod Entities** box have revised fields. These revised fields are marked *rev'd*. These fields are revisioned as described in “Revisioning Contact Information in Policies” on page 371. Revisioned fields implement the account syncable interface. For more information, see “Account Synchronization Classes for Contacts” on page 393.

See also

- “Policy Revisioning” on page 479
- “Extending the Contact Data Model” on page 125 in the *Contact Management Guide*

Linked Addresses Object Model

The following illustration shows some of the entities associated with contacts and linked addresses.



The **LinkedAddress** entity contains an array of linked addresses.

When you link an address to another address, PolicyCenter takes one of the following actions:

- If the other address is not already linked, PolicyCenter creates a new **LinkedAddress**.
- If the other address is already linked, PolicyCenter adds the address to the existing **LinkedAddress**.

You can link to an address on an **AccountContact** that has the role **NamedInsured** or **AccountHolder**. You can also link to an address on a contact that is the **PrimaryNamedInsured** on the **PolicyPeriod**.

See also

- “Linking an Address Between Multiple Contacts” on page 374
- “Working with Linked Addresses” on page 385
- “Configuring Linked Addresses for Contacts” on page 405

Contact Roles for Accounts and Policies

The PolicyCenter base application has contact roles defined at the account and policy levels. These roles are subtypes of the **AccountContactRole** and **PolicyContactRole** entities. The **AccountContactRole** entity represents a contact filling a role on the account, such as Maria Smith as a driver on the account. The **PolicyContactRole** represents a contact filling a role on the policy, such as Maria Smith as the primary named insured on the policy.

The following table lists the contact roles at the account level and the corresponding roles at the policy level.

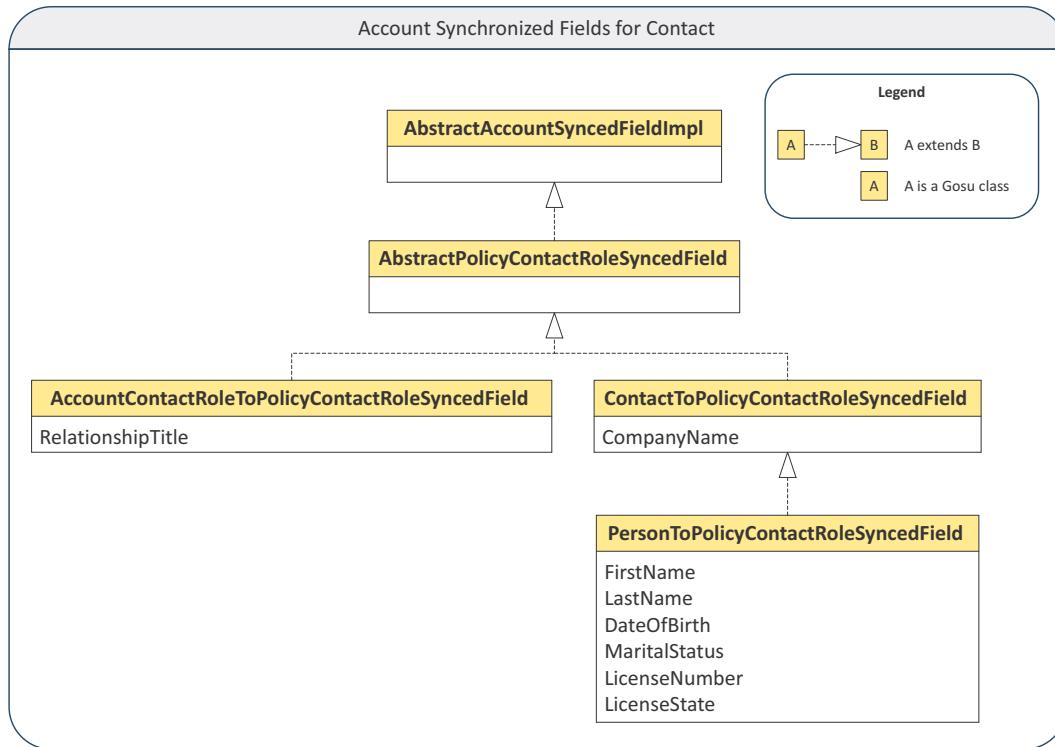
AccountContactRole subtype	PolicyContactRole subtype
AccountHolder	
AccountingContact	
SecondaryContact	
ClaimsInfoContact	
InspectionContact	
AccountingContact	
NamedInsured	PolicyNamedInsured
Driver	PolicyDriver
BillingContact	PolicyBillingContact
AdditionalInsured	PolicyAddlInsured
AdditionalInterest	PolicyAddlInterest
SuppliedEmployee	PolicySuppliedEmployee
ReceivedEmployee	PolicyReceivedEmployee
MiscContact	PolicyMiscContact
WCPolicyContactRole	
OwnerOfficer	PolicyOwnerOfficer (subtype of WCPolicyContactRole)
	WCLaborContact (subtype of WCPolicyContactRole)
LaborClient	PolicyLaborClient (subtype of WCLaborContact)
LaborContractor	PolicyLaborContractor (subtype of WCLaborContact)

The roles in the first column are subtypes of `AccountContactRole` which can be added to the account level. If there is no corresponding role in the `PolicyContactRole` column, the account contact role can only be associated with a contact added to the account. These roles represent people or company contacts with roles associated with the account but not with individual policies.

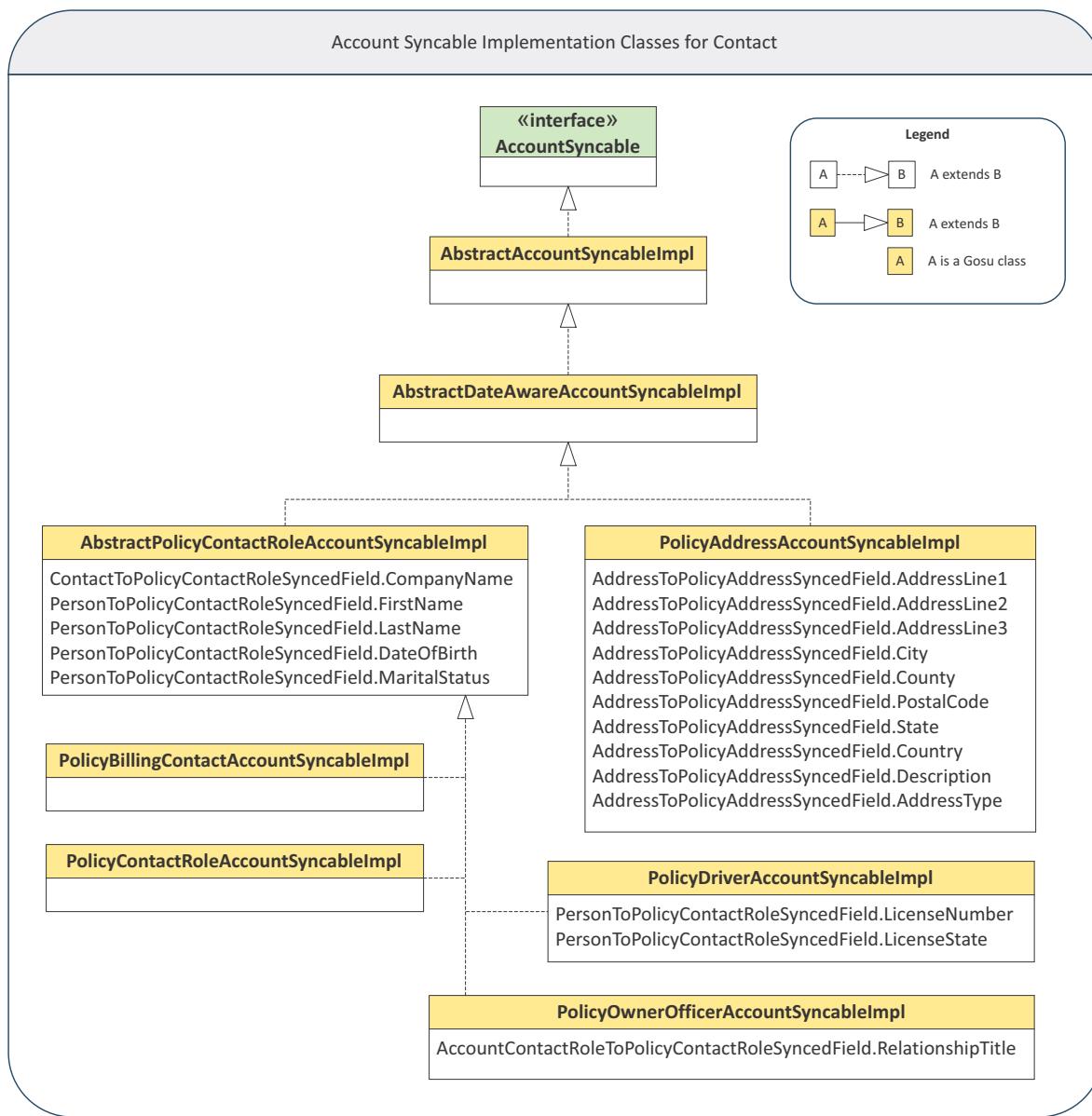
In some cases, a policy contact role is associated with an account level contact role, such as `PolicyDriver` and `Driver`. A policy contact role can be added to any of the policies within that account. (Many roles only apply to certain lines of business.) A contact role can contain fields that are shared across policies and other fields that are different across policies.

Account Synchronization Classes for Contacts

The following illustration shows the Gosu classes that implement synchronized fields on the account for contacts.



The following illustration shows how the `AccountSyncable` interface is extended for contacts.



See also

- “Adding a Revised Field to a Contact” on page 399
- “Synchronizing Contacts with Accounts” on page 582 in the *Integration Guide*

Configuring Contacts

There are multiple levels of contact configurability.

- `PolicyContactRole` and `AccountContactRole` are extendable, as are all of the roles in the default configuration.
- You can define new contact roles by defining new subtypes of `PolicyContactRole` and `AccountContactRole`. Accompanied by appropriate account and policy level user interfaces, the new contact roles are automatically integrated into the contact scheme of the application.

- You can configure which fields on a contact role are revised.
- You can define whether a particular contact role can be held by a Person, Company, or both.
- If you do not need them, you can disable contact roles defined in the default configuration.
- You can configure at what points and on what policy transactions PolicyCenter synchronizes contact information. This behavior can vary by role.

This topic includes:

- “Data Model Patterns for Contact Roles” on page 395
- “Adding New Contact Roles” on page 395
- “Adding a Revised Field to a Contact” on page 399
- “Configuring Linked Addresses for Contacts” on page 405
- “Gosu Classes for Contacts” on page 406
- “Plugins for Contacts” on page 407
- “Contact Batch Process” on page 407
- “Configuring the Contact Tab” on page 408

Data Model Patterns for Contact Roles

If you create a new policy contact role, you can base its configuration on an existing role. Choose the role that uses the appropriate data model pattern:

- **Normal** – Array connected to an entity, simple properties
- **Singleton** – One and only one connected to an entity
- **Simple Details** – One contact, array of detail entities
- **Join Details** – Details join contact to another entity

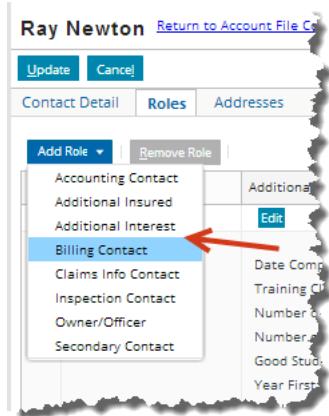
The following table shows the data model pattern used by some of the policy contact roles.

Policy contact role	Data model pattern	Attaches to
NamedInsured	Normal	Period
PrimaryNamedInsured	Singleton	Period
SecondaryNamedInsured	Singleton	PersonalAutoLine
LocationNamedInsured	Join Details	Period
AdditionalInsured	Simple Details	Line
BillingContact	Singleton	Period
Driver	Join Details	PersonalAutoLine
AdditionalInterest	Join Details	Period
OwnerOfficer	Normal	WorkersCompLine
LaborClient	Simple Details	WorkersCompLine
LaborContractor	Simple Details	WorkersCompLine

Adding New Contact Roles

In PolicyCenter, you can configure new contact roles by defining new subtypes of `PolicyContactRole` and `AccountContactRole`. The role appears automatically on the `AccountFile_Contacts` and `PolicyFile_Contacts` PCF pages. After you complete instructions to add the new contact role, `Audit Contact` will

appear in the list between Additional Interest and Billing Contact. You will be able to select this role on the Roles tab of the Contacts screen in the account and the policy.



The Audit Contact role will appear in the roles for that contact. In the following illustration, Audit Contact would appear between Account Holder and Driver.

Active	Name ↑	Role	Phone
Yes	Acta Motors	Named Insured	
Yes	BlackBURN's Plumbing	Named Insured	
Yes	EverReady Rentals	Named Insured	
Yes	Harry's Towing	Named Insured	
Yes	John Smith	Driver	650-333-33
Yes	Ray Newton	Account Holder, Driver, Named Insured	818-446-12
Yes	Speedy Glass Repair	Named Insured	

Roles ↑	Additional Information
Account Holder	Date Completed Training Class Training Class Type Number of Accidents 0 Number of Violations 0 Good Student Year First Licensed 1988 Qualifies for a Good Driver Discount No MVR Report Details
Driver	
Named Insured	

This topic provides guidance on adding an audit contact role and policy contact role, with step-by-step instructions for the audit contact role. To add a new contact role, you must:

- “Step 1: Create the Entities that Define the New Subtypes” on page 397
- “Step 2: Create an Implementation the Contact Configuration Plugin” on page 398
- “Step 3: Add Display Key and Entity Name” on page 398

See also

- “Using the PCF Editor” on page 285 in the *Configuration Guide*
- “The PolicyCenter Data Model” on page 149 in the *Configuration Guide*
- “Modifying the Base Data Model” on page 209 in the *Configuration Guide*

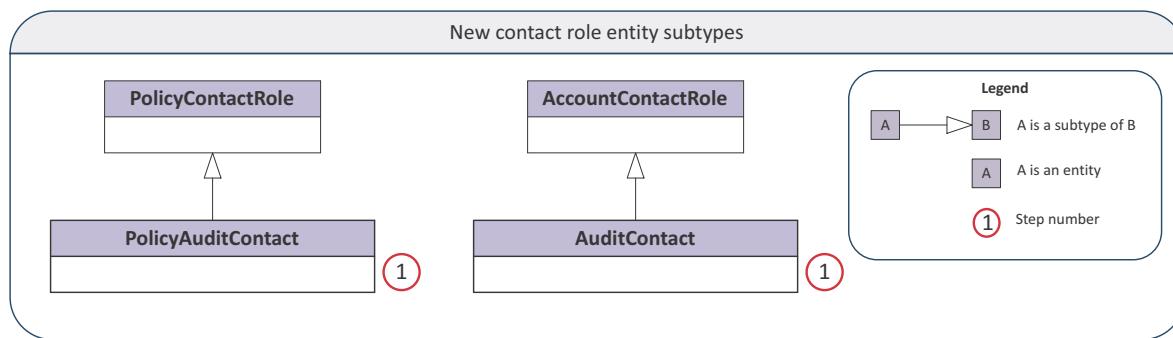
Step 1: Create the Entities that Define the New Subtypes

Create extension files for entities that define the new contact role subtypes. There must be subtypes for both the `AuditContactRole` at the account level (a subtype of `AccountContactRole`) and at the policy level (a subtype of `PolicyContactRole`).

- In this example, these contacts have no extra properties beyond the standard contact properties.
- If you need additional properties, define these in your subtype definition. Use the account contact subtype, `AuditContact`, to define any properties beyond the standard contact properties that you want to be the same across policies. Use the policy contact subtype, `PolicyAuditContact`, to define any properties that change across policies. In either case, the properties can change across policy revisions. If at any given time you want the value of the property to be different on two different policies, then configure the property at the policy level.
- If this is a new field, append `_Ext` to the field name avoid name conflicts with base PolicyCenter entities. See “Defining a New Data Entity” on page 212 in the *Configuration Guide*.

In this step, you create two new entities:

- A new `PolicyAuditContact` entity that is a subtype of `PolicyContactRole`.
- A new `AuditContact` entity that is a subtype of `AccountContactRole`.



Create the extension files for audit contact

1. In Studio, navigate to configuration → config → Extensions → Entity.
2. Right-click Entity, and choose New → Entity.
3. Enter the following information in the Entity dialog, then click OK:

Name	Value
Entity	<code>AuditContact_Ext</code>
Entity Type	subtype
Supertype	<code>AccountContactRole</code>

4. In `AuditContact_Ext.eti`, click subtype in the Element column to display the Name and Value columns for the entity. Enter the following value:

Name	Value
<code>displayName</code>	<code>AuditContact</code>

5. Add another entity named `PolicyAuditContact_Ext` that is a subtype of `PolicyContactRole`. Define the entity with the following values:

Name	Value
Entity	<code>PolicyAuditContact_Ext</code>
Entity Type	<code>subtype</code>
Desc	<code>Policy Audit Contact</code>
Supertype	<code>PolicyContactRole</code>
displayName	<code>PolicyAuditContact</code>

Note: If the new role is only applied at the account level, such as `AccountHolder`, then you do not need the second subtype, `PolicyAuditContact`.

Step 2: Create an Implementation the Contact Configuration Plugin

Create an implementation of the `IContactConfigPlugin` interface based on `ContactConfigPlugin.gs` in the `gw.plugin.contact.impl` package. Your implementation maps the new `AccountContactRole` subtype to the new `PolicyContactRole` subtype.

Implement contact configuration plugin

- In Studio, type CTRL+N, enter “`ContactConfigPlugin`”, and double-click to open `gw.plugin.contact.impl.ContactConfigPlugin`.
- Select all the text except for the package name in `ContactConfigPlugin.gs` and copy it to the clipboard.
- Navigate to the `gw.plugin.contact.impl` package in `configuration → gsrc`.
- Right-click `impl` and select `New Gosu Class`.
- In the `New Gosu Class` dialog, enter “`ContactConfigPlugin_mine`” in the `Name` field, then click `OK`.
- Remove the class declaration, then paste the contents of the clipboard at the end of the file.
- Change the class name from `ContactConfigPlugin` to `ContactConfigPlugin_mine`.
- Add the following code to `protected property get DefaultConfigs`:


```
new ContactConfig(true, {"company", "person"}, typekey.AccountContactRole.get("AuditContact_Ext"),
  {typekey.PolicyContactRole.get("PolicyAuditContact_Ext")})
```

 - The first argument indicates whether the contact type is enabled.
 - If you want the contact role to only be a person, then use `{"person"}`; or if only a company, use `{"company"}`.
 - The third and fourth arguments map `AuditContact` to `PolicyAuditContact`. If there are additional policy contact roles that map to `AuditContact`, add the roles to this line.
 - If you create a role that only exists at the account level, then omit the fourth argument.

Update the plugin registry

- In Studio, navigate to `configuration → config → Plugins → registry` and open `IContactConfigPlugin.gwp`.
- In `IContactConfigPlugin.gwp`, click the search button next to the `Gosu Class` field and search for the name of your plugin. If you are following this example, the plugin name is `ContactConfigPlugin_mine`.

Step 3: Add Display Key and Entity Name

Add a display key and entity name for the `AuditContact_Ext` entity you created. After completing these steps, you can add new contacts of your new role type at the account file.

To add a display key

1. In Studio, navigate to configuration → config → Localizations → *Lang* and double-click **display.properties** to open **Display Keys**.
2. Add the following line:
`entity.AuditContact_Ext = Audit Contact`

To add an entity name

Add an entity name for the **AuditContact_Ext** entity that is associated with the display key.

1. In Studio, right-click **EntityNames** and select **New Entity Name**.
2. Enter “**AuditContact_Ext**” in the **Entity** field and click **OK**.
Studio opens **AuditContact_Ext.en**.

3. In the editor, enter the following in the text field of the **Default** tab at the bottom of the screen. Do not enter this text in the **Type** text box.

```
displaykey.entity.AuditContact_Ext
```

To create a contact that uses the new contact role

1. Restart PolicyCenter.
2. In PolicyCenter, go to an account.
3. Click the **Contacts** in the left sidebar.
4. On the **Account File Contacts** screen, select **Create New Contact**.

The drop-down list contains **Audit Contact**.

Step 4: Modify PCF files and Gosu classes

Change PCF files to configure how users to set these roles on submissions. You also probably need to add some methods in the Gosu classes. Buttons on the PCF pages will use these methods to add and remove contacts.

How you change the PCF files and Gosu classes depends on what you want to do with the contacts. You can examine how the submission policy transaction adds contact roles and follow those patterns. See “Data Model Patterns for Contact Roles” on page 395.

Adding a Revised Field to a Contact

This topic describes how to add revised contact fields. Contacts have a more complex revisioning logic than locations. Contacts, as opposed to locations, have a distinct time window in which synchronization of revised fields occurs. If a change occurs after this window, a batch process applies the future-dated change to the revised field. If the change occurs before this window, a user decides when to apply the back-dated change to the revised field. For more information, see “Revisioning Contact Information in Policies” on page 371.

This topic provides general instructions and step-by-step instructions for adding a revised field to a contact. The step-by-step instructions guide you to add a **AuditLicenseNumber** revised field to the **AuditContact** entity. (For instructions on how to create the **AuditContact** and **PolicyAuditContact** entities, see “Adding New Contact Roles” on page 395.)

Follow these steps to add a revised field to **AuditContact**:

- “Step 1: Define the Revised Field on the Account Audit Contact” on page 400
- “Step 2: Define the Revised Field on the Policy Audit Contact” on page 400
- “Step 3: Define the Field as Syncable on the Policy Contact Role” on page 401
- “Step 4: Define the Field as Syncable on the Account Contact Role” on page 403

- “Step 5: Extend Entity and Gosu Class for Future-date Policy Changes” on page 404
- “Step 6: Add Get and Set Methods to the Policy Contact Role” on page 404
- “Step 7: Add the Revisioned Field to PolicyCenter User Interface” on page 405

See also

- “The PolicyCenter Data Model” on page 149 in the *Configuration Guide*
- “Modifying the Base Data Model” on page 209 in the *Configuration Guide*
- “Extending a Base Configuration Entity” on page 213 in the *Configuration Guide*

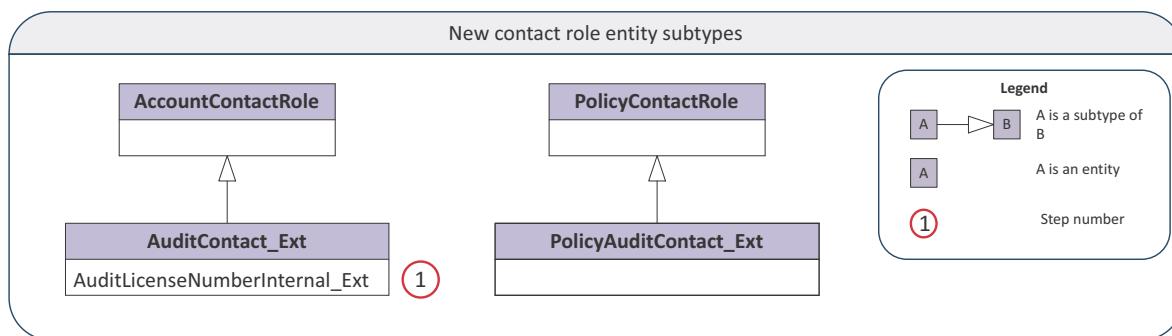
Step 1: Define the Revisioned Field on the Account Audit Contact

Define the field on the account. In this example, the revisioned field is on the `AuditContact` entity. If the field already exists, you do not need to define it. If the field does not exist, add the field to an extension entity for `AuditContact`.

The step-by-step instructions add an audit license number field to the audit contact entity. Use the following naming convention:

- Append `Internal` to the field name to signal an internal field. This convention is a Guidewire best practice for internal field names. All revisioned account syncable fields in the base configuration follow this naming convention.
 - If this is a new field, append `_Ext` to the field name avoid name conflicts with base PolicyCenter entities.
- See “Defining a New Data Entity” on page 212 in the *Configuration Guide*.

In the following illustration, the circled number 1 shows the `AuditLicenseNumberInternal_Ext` field that you will create on the `AuditContact_Ext` entity.



To add the revisioned field to the account audit contact

Define the revisioned field for `AuditContact_Ext`.

- In Studio, navigate to `configuration` → `config` → `Extensions` → `Entity` and double-click to open `AuditContact_Ext.eti`.
- Next to `+`, click the drop-down list and choose `column`.
- Enter the following values for the new column which define an account level field, `AuditLicenseNumber`:

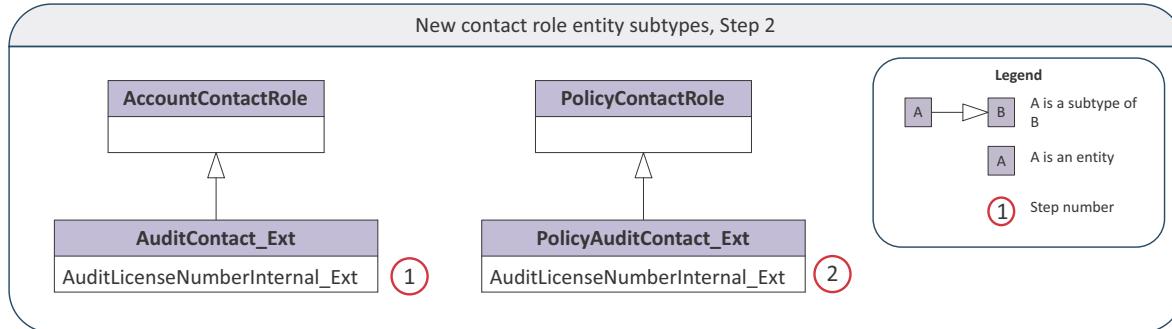
Name	Value
<code>name</code>	<code>AuditLicenseNumberInternal_Ext</code>
<code>type</code>	<code>mediumtext</code>
<code>desc</code>	The audit license number

Step 2: Define the Revisioned Field on the Policy Audit Contact

Define an internal field on the `PolicyAuditContact_Ext` entity. Define the field as follows:

- Set the value of the `setterScriptability` and `getterScriptability` attributes to `doesNotExist`. This value makes the field internal and not visible to Gosu code. The field will not appear in the data dictionary. For more information about these attributes, see “`<column>`” on page 180 in the *Configuration Guide*.
- Define the `PolicyAuditContact_Ext` entity as implementing the `AccountSyncable` interface.

In the following illustration, the circled number 2 shows the `AuditLicenseNumberInternal_Ext` field that you will create on the `PolicyAuditContact` entity.



To define the revised field on the policy audit contact

- In Studio, navigate to `PolicyAuditContact_ext.eti` entity as you did in “Step 1: Define the Revised Field on the Account Audit Contact” on page 400.
- Open `PolicyAuditContact_Ext.eti`.
- Next to `+`, click the drop-down list and choose `column`.
- Enter the following values for the new column:

Name	Value
<code>name</code>	<code>AuditLicenseNumberInternal_Ext</code>
<code>type</code>	<code>mediumtext</code>
<code>desc</code>	The policy audit contact
<code>getterScriptability</code>	<code>doesNotExist</code>
<code>setterScriptability</code>	<code>doesNotExist</code>

- Click subtype at the top of the `Element` hierarchy.
- Next to `+`, click the drop-down list and choose `implementsInterface`.
- Enter the following values for `implementsInterface`:

Name	Value
<code>iface</code>	<code>gw.api.domain.account.AccountSyncable</code>
<code>impl</code>	<code>gw.contact.PolicyAuditContactAccountSyncableImpl</code>

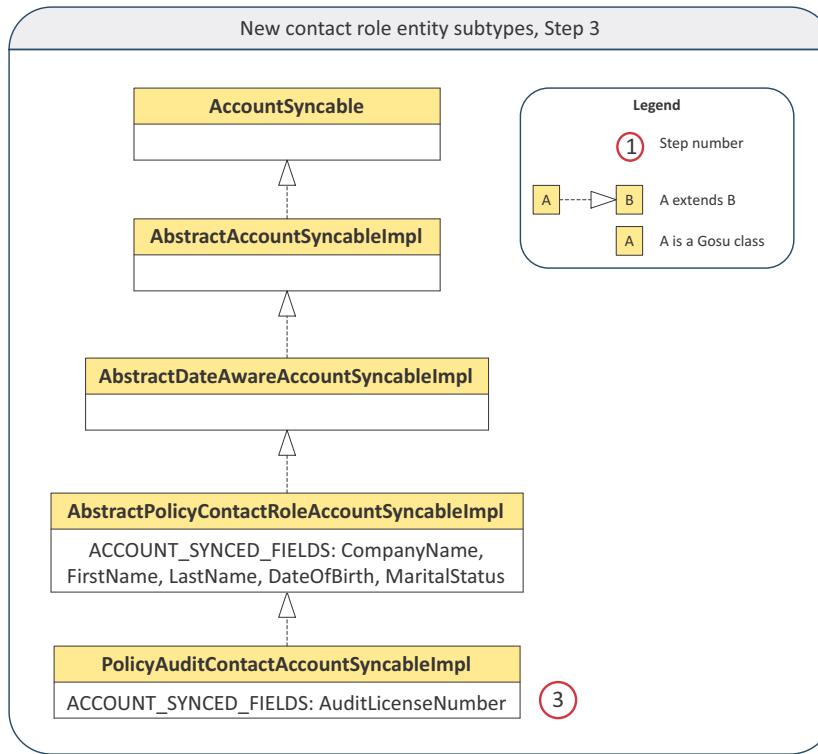
Studio displays a message that the implementation does not exist. You add this implementation in later steps.

Step 3: Define the Field as Syncable on the Policy Contact Role

Define the field as syncable on the policy contact role by adding it to the `ACCOUNT_SYNCED_FIELDS` variable in the `gw.contact.PolicyAuditContactAccountSyncableImpl` Gosu class.

Because this is a new contact role, you must first create the account syncable implementation class. The code is based on the class for the policy driver contact role: `gw.lob.pa.contact.PolicyDriverAccountSyncableImpl`.

In the following illustration, the circled number 3 shows the `ACCOUNT_SYNCED_FIELDS` variable on the new `PolicyAuditContactAccountSyncableImpl` class. Because all revised contact information is effective dated, all revised fields on contact must extend the `AbstractDateAwareAccountSyncableImpl` class. For more information, see “Revisioning Contact Information in Policies” on page 371.



In this class, when you get the `AccountSyncedFields` property, the code returns an array. The array includes the names of the synchronized fields in the current class and in all syncable classes that it extends. Therefore, when you get the `AccountSyncedFields` property, the array contains `AuditLicenseNumber`, `CompanyName`, `FirstName`, `LastName`, `DateOfBirth`, and `MaritalStatus`.

To define the account syncable implementation class

1. In Studio, navigate to the `gw.contact` package.
2. Right-click `contact`, select `New → Gosu Class` and enter “`PolicyAuditContactAccountSyncableImpl`” for the class name and click `OK`.

To define the revised field as syncable on the policy contact role

1. In Studio, open `gw.contact.PolicyAuditContactAccountSyncableImpl.gs`.
2. Add the following code which is based on `PolicyDriverAccountSyncableImpl.gs`.

```

uses gw.contact.AbstractPolicyContactRoleAccountSyncableImpl
uses gw.account.AccountContactRoleToPolicyContactRoleSyncedField
uses com.google.common.collect.ImmutableSet
uses gw.api.domain.account.AccountSyncedField
uses gw.api.domain.account.AccountSyncable
uses java.util.Set

/**
 * Implementation that handles AuditContact account syncing behavior.
 */
@Export
class PolicyAuditContactAccountSyncableImpl extends
  
```

```

AbstractPolicyContactRoleAccountSyncableImpl<PolicyAuditContact_Ext> {

    static final var ACCOUNT_SYNCED_FIELDS = ImmutableSet.copyOf(
        (
            AbstractPolicyContactRoleAccountSyncableImpl.AccountSyncedFieldsInternal.union(
                {
                    AccountContactRoleToPolicyContactRoleSyncedField.AuditLicenseNumber
                }
            )
        )
    )

    protected static property get AccountSyncedFieldsInternal() :
        Set<AccountSyncedField<AccountSyncable, ?>> { // provided so subclasses can extend this list
    return ACCOUNT_SYNCED_FIELDS
}

construct(accountSyncable : PolicyAuditContact_Ext) {
    super(accountSyncable)
}

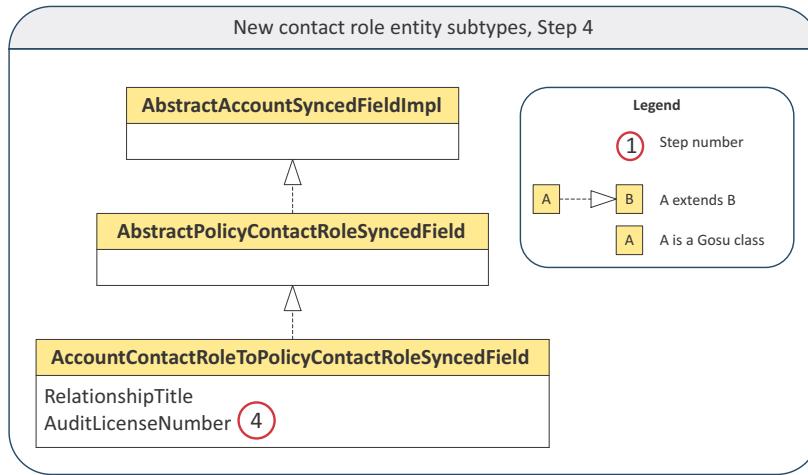
override property get AccountSyncedFields() : Set<AccountSyncedField<AccountSyncable, ?>> {
    // must override to ensure that we call the correct static AccountSyncedFieldsInternal property
    return AccountSyncedFieldsInternal
}
}

```

Step 4: Define the Field as Syncable on the Account Contact Role

Add the new field to the `gw.account.AccountContactRoleToPolicyContactRoleSyncedField` class. Define the new field as revised and syncable on the account.

In the following illustration, the circled number 4 shows the `AuditLicenseNumber` variable definition as a synchronized field.



The `AccountContactRoleToPolicyContactRoleSyncedField` constructor assumes that the field name is appended by `Internal`. If the field name is appended by `Internal`, copy one of the existing the variable definitions and modify it for the new field. For example, you can copy and modify `RelationshipTitle`.

If the field name is appended by `Internal` and an extension, `Internal_Ext` for example, copy one of the existing the variable definitions and modify it for the new field. Then add a second parameter for the name of the field on the policy location. In addition, define a constructor with arguments for the `accountEntityFieldName` method.

The instructions describe how to define the field if it is appended by `Internal_Ext`.

To define the revised field appended with `Internal_Ext`

1. In Studio, open `gw.account.AccountContactRoleToPolicyContactRoleSyncedField.gs`.

2. Add the `AuditLicenseNumber` field to the class:

```

class AccountContactRoleToPolicyContactRoleSyncedField<S extends PolicyContactRole, T>
extends AbstractPolicyContactRoleSyncedField<String> {

```

```

...
public static final var AuditLicenseNumber :
    AccountContactRoleToPolicyContactRoleSyncedField<PolicyAuditContact_Ext, String> =
        new AccountContactRoleToPolicyContactRoleSyncedField<PolicyAuditContact_Ext, String>
            ("AuditLicenseNumber", "AuditLicenseNumberInternal_Ext")
...

```

3. Define a constructor for accountEntityFieldName with the following arguments:

```

construct(accountEntityFieldNameArg : String, policyEntityFieldNameArg : String) {
    super(accountEntityFieldNameArg, policyEntityFieldNameArg, accountEntityFieldNameArg + "_Ext",
        accountEntityFieldNameArg + "IsNull_Ext", PendingAccountContactRoleUpdate)
}

```

The last two parameters are for future dated policy changes to the contact name. These changes will be applied to the account at that future date.

Step 5: Extend Entity and Gosu Class for Future-date Policy Changes

In certain circumstances, a policy may have a change to the contact name that takes effect at a future date. This change will be applied the account contact at that future date. For more information, see “Revisioning Contact Information in Policies” on page 371.

Because of future-dated changes, you must extend the PendingAccountContactRoleUpdate entity. The PendingEntityUpdate entities are necessary to update the account level fields when PolicyCenter changes those fields on the policy in the future.

1. Navigate to configuration → config → Extensions → Entity.
2. Right-click Entity and select New → Entity Extension.
3. Enter “PendingAccountContactRoleUpdate” in the text box and click OK.
4. In the extension file, add a column with the following values:

Name	Value
name	AuditLicenseNumber_Ext
type	mediumtext
desc	The audit license number that needs to be updated on the AccountContactRole at a future date.

5. Add another column with the following values:

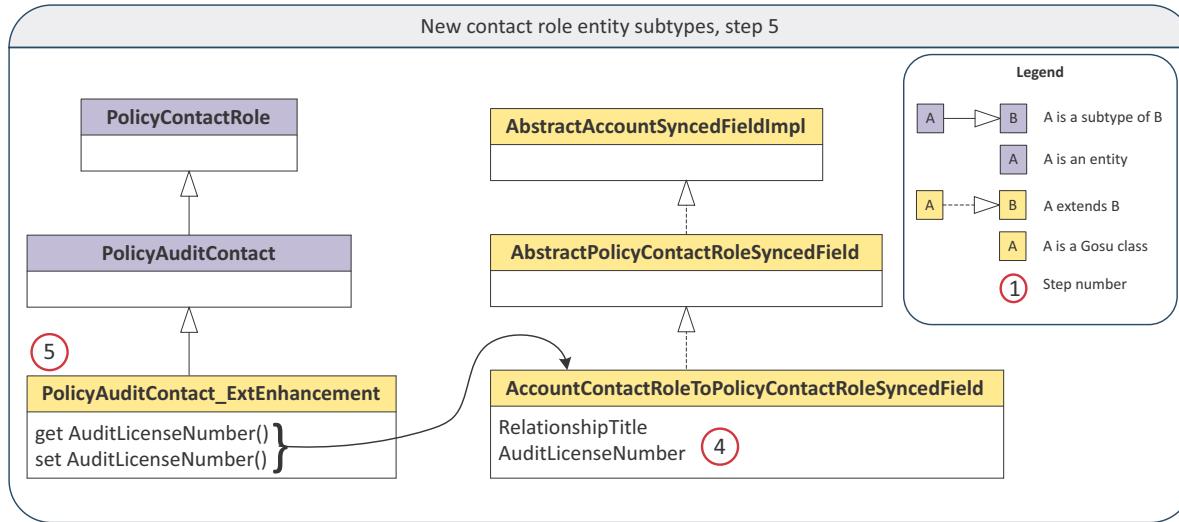
Name	Value
name	AuditLicenseNumberIsNull_Ext
type	bit
default	false
desc	True if the AuditLicenseNumber_Ext field will be set to null.

6. In gw.account.PendingAccountContactRoleUpdateAdapter.gs, add code to the applyUpdate method that copies these two properties.

Step 6: Add Get and Set Methods to the Policy Contact Role

Since you created a new PolicyAuditContact_Ext contact role, you must first create an enhancement.

Add get and set methods for syncable fields to `gw.contact.PolicyAuditContact_ExtEnhancement`. The get and set method access the `AccountContactRoleToPolicyContactRoleSyncedField` class. In the following illustration, the circled number 5 shows the get and set methods on the `PolicyAuditContact_ExtEnhancement` class.



To create a new enhancement for the contact role

1. In Studio, navigate to the `gw.contact` package.
 2. Right-click `contact` and select `New → Gosu Enhancement`.
 3. In the `New Gosu Enhancement` dialog, enter “`PolicyAuditContact_ExtEnhancement`” in `Name` and “`PolicyAuditContact_Ext`” in `Enhanced type`, and click `OK`.
- Studio creates `gw.contact.PolicyAuditContact_ExtEnhancement.gsx`.

To add get and set methods to the policy location

1. In Studio, open `gw.policylocation.PolicyAuditContact_ExtEnhancement.gsx`.

2. Underneath the package statement, add:

```
uses gw.account.AccountContactRoleToPolicyContactRoleSyncedField
```

3. In the enhancement, insert get and set methods for `AuditLicenseNumber`:

```
/***
 * Shared and revised AuditLicenseNumber.
 */
property get AuditLicenseNumber() : String {
    return AccountContactRoleToPolicyContactRoleSyncedField.AuditLicenseNumber.getValue(this)
}

/***
 * Shared and revised AuditLicenseNumber.
 */
property set AuditLicenseNumber(arg : String) {
    AccountContactRoleToPolicyContactRoleSyncedField.AuditLicenseNumber.setValue(this, arg)
}
```

Step 7: Add the Revised Field to PolicyCenter User Interface

To see this field in PolicyCenter, you must add it to various PCF pages.

Configuring Linked Addresses for Contacts

In the default configuration, you can link to addresses in the following types of contacts:

- `PrimaryNamedInsured` on `PolicyPeriod`

- `AccountHolder` on `Account`
- `NamedInsured` on `Account`

The `gw.address.LinkedAddressUIHelper` class contains methods related to linked addresses.

The `getContactsAvailableAsLinks` method creates a list of contacts with linkable addresses. To build this list, the method calls `getPeriodContacts` and `getAccountContacts`.

The `getPeriodContacts` method returns the list of contacts on the policy period that have linkable addresses. In the default configuration, this method returns the primary named insured, unless the primary named insured is the current contact. Modify this method if you need to link to addresses on other types of contacts on the policy or related to revised data.

The `getAccountContacts` method returns the list of contacts on the account that have linkable addresses. In the default configuration, this method returns the account holder and named insureds. Modify this method if you need to link to addresses on other types of contacts on the account.

The `LinkAddressInputSet` PCF file displays the list of addresses for contacts in the following order:

1. Account holder
2. Primary named insured
3. Name insureds

You can specify a priority order when you add a contact to the list. Within each type of contact, the `LinkAddressInputSet` PCF file displays the list of addresses in priority order starting at 1.

Linked Address API

The `Address` class provides the following methods for working with linked addresses:

- `linkAddress`
- `unlink`
- `updateLinkedAddresses`
- `isLinkedToAddress`

As always, use the `safeRemoveAddress` method in `gw.contact.ContactEnhancement` when you remove an address. This method safely removes a link to another address.

See also

- “Linking an Address Between Multiple Contacts” on page 374
- “Working with Linked Addresses” on page 385
- “Linked Addresses Object Model” on page 391
- “Address APIs” on page 589 in the *Integration Guide*

Gosu Classes for Contacts

You can find the Gosu classes for Contact by navigating to `configuration → gsrc` and finding the `Classes.gw.contact` package in Studio.

Adapters and Enhancements for Revised Fields

Policy period entities with revised fields have adapters and enhancements that maintain and synchronize these fields. Before the policy is bound, these fields get their value from the account. At binding, these fields are copied to properties in a Gosu enhancement associated with the entity. The adapters update the fields in the entity by calling the getter and setter methods in the enhancement.

The following table lists the enhancements and adapters provided in the base configuration.

Files	Fields
PolicyContactRoleSyncableAdapter.gs PolicyContactRoleEnhancement.gsx	Synchronizes CompanyName, FirstName, and LastName for all policy contact roles. These files are in the gw.contact package.
PolicyDriverSyncableAdapter.gs PolicyDriverEnhancement.gs	Synchronizes LicenseNumber and LicenseState for PolicyDriver. This role is specific to personal auto. These files are in the gw.lob.pa.contact package.
PolicyOwnerOfficerSyncableAdapter.gs PolicyOwnerOfficerEnhancement.gs	Synchronizes RelationshipTitle for PolicyOwnerOfficer. This role is specific to workers' compensation. These files are in the gw.lob.wc.contact package.

Plugins for Contacts

The following plugins are related to contacts.

Contact Configuration Plugin

The `IContactConfigPlugin` configures the `Contact` entity.

The `IContactConfigPlugin` maps the `PolicyContactRole` to the corresponding `AccountContactRole`. For example, the code maps `PolicyNamedInsured` to `NamedInsured`.

The code also controls which `Contact` subtypes (Person or Company) are allowed for each role. For example, it specifies that an `AccountHolder` can be a Person or a Company, but a `Driver` or a `PolicyDriver` can only be a Person.

The `IContactConfigPlugin` can also disable roles. You disable roles by setting first argument in the line configuring the contact to `false`:

```
new ContactConfig(false, {"company", "person"}, "AuditContact", {"PolicyAuditContact"}),
```

In the default configuration, each `PolicyContactRole` references one `AccountContactRole`. However, you can configure several `PolicyContactRoles` to reference one `AccountContactRole`.

Account Syncable Plugin

The `AccountSyncablePlugin` implements the `IAccountSyncablePlugin` interface. This plugin implementation handles the synchronization of revised fields between accounts and other entities such as contacts and locations. You can view this plugin by navigating to `Classes` → `gw` → `plugin` → `account` → `impl` in Studio.

The only method is `refreshAccountInformation`, which refreshes account information to ensure the account has the most current data when calling the other methods. This method takes an entity instance that implements the main `AccountSyncable` interface.

See also

- “Revisioning Contact Information in Policies” on page 371
- “Synchronizing Contacts with Accounts” on page 582 in the *Integration Guide*

Contact Batch Process

The `Apply Pending Account Data Updates` batch process applies pending updates to account data such as an update to revised contact information. In the default configuration, this batch process runs every day at 12:10 a.m.

For example, if you make a future-dated policy change to a policy contact's name, the batch process applies the name change to the account on the policy change effective date.

See also

- “Revisioning Contact Information in Policies” on page 371
- “List of Batch Processes and Distributable Work Queues” on page 97 in the *System Administration Guide*

Configuring the Contact Tab

This topic describes how to configure the **Contact** tab.

See also

- “Centralized View of Contacts on the Contact Tab” on page 370
- “Working with the Contact Tab” on page 374

Gosu Classes that Implement Features for the Contact Tab

The following table describes some of the classes that implement features related to the **Contact** tab.

Gosu class	Description
ContactAssociationFinder.gs	Contains code for finding accounts, policies, claims, and billing accounts related to the contact file.
ProductCodeFilterSet.gs	Contains code for filtering products on contact file list views.
JobFilters.gs	Contains code for filtering status and policy transactions in contact file list views. See the StatusFilterSet and JobTypeFilterSet methods.
BillingAccountInfo.gs	Contains code that builds the BillingAccountInfo object. Fields in this object appear in the Contact File Billing screen.
PolicyPeriodBaseEnhancement.gsx	Contains code that gets the PeriodDisplayStatus. The Contact File Policies screen uses this status to filter policy periods.
ClaimSearchCriteria.gs	Contains code for searching on the Contact File Claims screen.
ContactEnhancement.gsx	Contains code that gets the PolicyPeriods property. The Contact File Claims screen uses this property to return the bound policy periods that are related to this contact and for which the user has view permissions.

PCF Files in the Contact Tab User Interface

The following table describes some of the PCF files used in the **Contact** tab user interface.

PCF file	Description
TabBar.pcf	Contains the Contact tab.
ContactFile.pcf	The main PCF file for displaying Contact File Details . The default tab is ContactFile_Details.pcf.
ContactFile_Accounts.pcf	The main PCF file for Contact File Accounts .
ContactFile_Billing.pcf	The main PCF file for Contact File Billing .
BCAccount.pcf	Exit point to BillingCenter .
ContactFile_Claims.pcf	The main PCF file for Contact File Claims .
ContactClaimsLV.pcf	Contact claims search results list view.
ClaimDetailsCV.pcf	Detail card panel for selected claim in search results list view.
ClaimDetailsDV.pcf	Detail view for selected claim in search results list view.
ViewClaim.pcf	Exit point to ClaimCenter .
ContactFile_Details.pcf	The main PCF file that contains the Contact Detail and Addresses tabs on the Contact File Details screen. The PCF files for each tab are AccountContactDV.pcf and AddressesPanelSet.pcf.
ContactFile_Policies.pcf	The main PCF file for the Contact File Policies page.

PCF file	Description
ContactFile_WorkOrders.pcf	The main PCF file for Contact File Policy Transactions.
ContactFileInfoBar.pcf	Displays brief information about the contact. Appears below the Contact tab and above the Contact File.
ContactFileMenuActions.pcf	The Actions menu for the contact file.
CreateAccountForContact.pcf	Account creation page for the contact file.
ContactForward.pcf	Forwards the user to Search Contacts or the contact file depending on the contact.
ExternalContactFile.pcf	Displays the contact file for external contacts.
ExternalContactFile_Details.pcf	Displays the Contact Detail and Addresses tabs for external contacts. The PCF files for each tab are AccountContactDV.pcf and AddressesPanelSet.pcf.
ExternalContactFileMenuActions.pcf	The Actions menu for the contact file for an external contact.
NewContact.pcf	The main PCF file for the New Contact page.
ContactPanelSet.pcf	Main panel set for the create new contact page.
ContactSearch.pcf	Main PCF file for Search Contacts page.
ContactSearchScreen.pcf	Main screen for contact search which displays search parameter and search results.
UserPrefsDV.pcf	User preferences page, which includes the maximum recent contacts setting for each user.

Underwriting Authority

Underwriting authority in PolicyCenter provides an extensible underwriting infrastructure and example rules and practices which you can modify or augment. The underwriting rules specify such things as jurisdictions of coverage and minimum or maximum amounts. For example, when an agent creates a new policy, certain terms of that policy may need underwriting review before the policy can be bound. The underwriting rules of the carrier require that an underwriter approve any vehicle valued over \$100,000. If an agent adds a car valued at \$200,000 in a personal auto policy, an underwriter must give approval for that car. PolicyCenter creates an issue for the underwriter, and the agent cannot bind the policy until the underwriter approves the issue.

This topic includes:

- “Underwriting Authority Overview” on page 411
- “Working with Underwriting Issues” on page 414
- “Working with Underwriting Referral Reasons” on page 416

See also

- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*
- “Authority Profiles” on page 655

Underwriting Authority Overview

Underwriting authority in PolicyCenter provides the following features.

Issue types specify the different kinds of underwriting-related issues. Issue types specify the point at which policy transactions are stopped. One or more *authority profiles* are assigned to individual users. These profiles allow the user to approve issues within the levels specified in the *authority grants* for each specific issue. Issue types also specify related user interface details and defaults.

The default configuration contains authority profiles for agents, underwriters, and an underwriting manager. You assign authority profiles to each user.

As part of the underwriting process, agents can pass policies to underwriters to obtain approval for issues. In the default configuration, when a policy is passed to an underwriter, the agent can no longer edit it. The policy is in

the **Under UW Review** state. If the policy has an issue that blocks quoting, the agent cannot view the quote until the underwriter approves the issue and releases the policy back to the agent. The agent can view the quote, then bind and issue the policy. Or the agent can edit the policy, potentially raising new issues.

In addition to the agent requesting approval for issues, in the default configuration there are two other ways that a policy commonly can enter the **Under UW Review** state. A policy can enter the **Under UW Review** state when an underwriter takes a policy by clicking the **Lock for Review** button. A policy can also enter the **Under UW Review** state when a user with the `editlockoverride` permission quotes a policy that is not already locked. For example, the user receives a call from an agent to approve an issue, rather than receiving a request for approval in PolicyCenter.

PolicyCenter raises issues automatically based on policy choices such as the types of vehicles on a policy or coverage amounts. The user can also raise issues manually by adding them to the policy. External systems can also use an API to add underwriting referral reasons to a policy. Underwriting referral reasons cause underwriting issues to be raised the next time a policy transaction is run on that policy.

Specifying Values

PolicyCenter can raise an issue that has a value associated with it. These *value issues* indicate a sense of how risky, or severe an issue is. The value can come from several places. The value can be copied from the policy, such as the replacement value of a vehicle. It can be a simple calculation, such as the number of days a requested change is post-dated. It can be a more complex calculation that takes into account many factors. Values can also be non-numeric, such as the set of jurisdictions in which a user can write a policy.

You can specify a minimum or maximum value that causes an issue to be created. For example, you can raise an issue for cars valued over \$40,000. In addition, for each type of user, you can define approval limits. Therefore, agents can approve cars valued up to \$50,000, but underwriters can approve cars valued up to \$100,000.

You can also specify inclusion in a set of jurisdictions. For example, underwriters who writes policies for the western region can only approve cars garaged in California, Oregon, and Washington.

For value issues that specify a minimum or maximum value, you can specify whether the default approval limit is for the exact value. You can also specify whether to add an additional percentage or amount to the value.

You can configure PolicyCenter to handle additional values types or sets.

Checking Sets and Blocking Points

You can define checking sets and blocking points for each issue.

Checking sets are the places in a policy transaction where PolicyCenter checks to see if an issue needs to be raised. You specify a checking set value for each issue type. For example, PolicyCenter checks for high-valued vehicles before quoting the policy.

Blocking points stop progress of the policy transaction until an issue is approved. You define a blocking point for each issue type. For example, PolicyCenter prevents a user from binding a policy until high-value vehicle issues are approved.

For more information, see `CheckingSet` and `BlockingPoint` in “Underwriting Issue Type System Table” on page 467 in the *Configuration Guide*.

Underwriting Referral Reasons on the Policy

Underwriting referral reasons are usually only used when a notable condition arises outside of a policy transaction, possibly outside of the data that PolicyCenter maintains on the policy. If the condition can be evaluated based on data in PolicyCenter, then it is usually preferable to use an underwriting issue rather than an underwriting referral reason. You can add underwriting referral reasons manually. External systems can also add underwriting referral reasons through an API.

For example, an external claim system stores the loss claims for a policy. The external system notes that loss claims are unacceptably high. The external system sends a message to PolicyCenter to add an underwriting referral reason on the policy. So the next time a policy transaction runs on the policy, an issue is created that corresponds to this referral reason. The referral reason remains on the policy.

You cannot approve an underwriting referral reason, but you can remove it. For example, a visit to the customer site revealed numerous safety violations, and the underwriter added an underwriting referral reason to the policy. On a subsequent visit, the underwriter finds that the safety violations have been addressed. The underwriter removes underwriting referral reason from the policy. The next time a policy transaction is run or reaches a checking set, the corresponding issue is deactivated.

Automatic Approvals

You can specify that PolicyCenter will approve issues of a specific type automatically if the current user has the correct authority. These issues are *auto-approvable* issues.

Auto-approvable issues provide automatic approval of issues that arise during normal policy transaction activities, so that the user does not have to approve the issue every time it occurs. An auto-approvable issue can be the general authority to write policies under a certain premium level or in a given jurisdiction.

When auto-approvable issues are automatically approved, the issues are approved to the default levels specified for that issue type. Auto-approvable issues are not automatically approved if there is a manual approval on the issue.

Automated Renewals

For automated renewal policy transactions, the default user for automatic approvals is `renewal_daemon`. In automated renewals, all issues are treated as auto-approvable issues.

Automated Policy Changes

For automated policy changes, `policychange_daemon` is the default user for automatic approvals. In automated policy changes, all issues are treated as auto-approvable issues.

Approval Duration

Certain user actions in the current or future policy transactions can invalidate an approval. For each issue, you can specify whether to remove an approval if the user edits the policy before binding.

You can specify the default duration of an approval. For example, an approval can remain in effect until the next change, until the end of the term, for one year, for three years, or until rescinded. For each issue type, you can configure the default duration. When approving the issue, you can change the duration. You can also add your own durations.

History of Underwriting Decisions

PolicyCenter maintains a history for each issue. PolicyCenter displays the history on the **Risk Approval Details** screen. The history includes details such as:

- The approver
- The date of the approval
- The effective date
- The point in the policy transaction through which the approval is valid
- When the approval expires

The history also includes a reference to the policy period to which the history event applies. The policy period may or may not be bound.

When multicurrency display is enabled, the **Risk Approval Details** screen has a **Currency** column. When approving a monetary underwriting issue, you can edit the amount, but cannot change the currency. The **Currency** field is preset to the approval value currency on the underwriting issue.

Working with Underwriting Issues

This topic describes how to work with underwriting issues in the PolicyCenter user interface.

UW Issues on the Risk Analysis Screen

PolicyCenter displays issues on the **Risk Analysis** screen, **Underwriting Issues** tab.

UW Issues Buttons

PolicyCenter displays the following buttons at the top of the screen:

Button	Description
Add UW Issue	Add an underwriting issue.
Request Approval	Request approval from another user, such as an underwriter. Takes you to the UW Activity screen where you create an activity to review and approve issues on the policy. The UW Activity screen allows you to select how you would like PolicyCenter to assign the activity.
Lock for Review	Lock the policy for underwriting review. The policy cannot be edited until you release the lock. This choice appears if you have the <code>editlockoverride</code> permission. After you click this button, all users see Under UW Review in the Info Bar .

View Issues Blocking

Use the **View Issues Blocking** pull-down list to filter issues as follows:

- **Me** – View all issues blocking the current user.
- **View All** – View all blocking issues.

PolicyCenter adds other user names to the drop-down list for the following policy transaction roles, if the roles are defined:

- **Initial Referrer** – The user whose actions first sent the policy into underwriting review.
- **Creator, Requestor, Processor** – The user who has this role.

If a user serves more than one role on the policy, that user appears only once.

Note: So than an agent cannot easily infer the authority levels of the underwriter, the drop-down list does not include the underwriter.

This list is controlled by the `getUsersWithUWPerspectives` method in `gw.assignment.JobAssignmentEnhancement.gsx`.

Issue Groups

Issues are grouped as follows:

- **Already Rejected** – All rejected issues.
- **By blocking type** – For example, all issues **Blocking Bind** or **Blocking Quote**. You can **Approve**, **Reject**, or **Reopen** a blocking issue.

- **Already Approved** – All approved issues.

Approval Buttons

PolicyCenter displays the following buttons for each issue:

Button	Description
Approve	Takes you to the Risk Approval Details screen.
Reject	This button appears for issues that you can reject. This button appears for all issues if you have the <code>uwreject</code> permission. A rejected issue prevents the policy transaction from crossing any blocking point.
Reopen	Reopen a previously approved or rejected issue. This button appears for issues that you approved or rejected. This button appears for all issues if you have the <code>uwreopen</code> permission.

Symbols

The symbols after the issue name represent the following:

- * The issue is an auto-approvable issue.
- § The issue varies over the policy period.
- (§) The issue varies over the policy period but does not block.

See also

- “Configuring Approvals” on page 473 in the *Configuration Guide*

Risk Approval Details Screen

When you click **Approve** on the **Risk Analysis** screen, PolicyCenter displays the **Risk Approval Details** screen. The default values for the approval are defined in the `UWIssueType` system table in Product Designer. The Reference

Value includes any default value offset or percentage. You can only approve values that are within your authority grant. This screen allows you to make changes to the approval. It also displays the approval history.

Field	Description
Allow Edit	Initial value of this radio button comes from DefaultEditBeforeBind column in the UWIssueType system table. Values are Yes and No. If Yes, the approval remains if the policy is edited before binding. If No, the approval is removed if the policy is edited before binding this policy transaction.
Through	The initial value for this drop-down list comes from the DefaultApprovalBlockingPoint column in the UWIssueType table. Values are: <ul style="list-style-type: none"> • Quote – Approval through quoting the policy. • Quote Release – Approval through releasing the quote on the policy. • Bind – Approval through binding the policy. • Issuance – Approval through issuing the policy.
Invalid from	The initial value for this drop-down list is from DefaultDurationType column in UWIssueType system table. Values are: <ul style="list-style-type: none"> • Next Change – The next policy change. • End of Term – The end of the policy term. • One Year – One year (minus one day) from the effective date of this policy transaction. • Three Years – Three years (minus one day) from the effective date of this policy transaction. • Rescinded – The approval is rescinded.
Offset Percentage	For value issues, if the DefaultValueAssignmentType column in the UWIssueType system table is set to Offset Percentage, then the DefaultValueOffsetAmount column value is used. The default approval amount in Reference Value field is for the value of the item plus (or minus) the offset percentage. For example, the offset percentage for a High Value Vehicle is 10 and the comparator is At most. The system triggers an issue if the value is greater than \$100,000. If the value of the vehicle is \$300,000, then the default approval amount is \$330,000.
Offset Amount	If the DefaultValueAssignmentType column in the UWIssueType system table is set to Offset Amount, then the DefaultValueOffsetAmount column value is used. The default approval amount in the Reference Value field is for the number of the item plus (or minus) the offset amount. For example, the offset amount for Number of Vehicles is 1 and the comparator is At most. The system triggers an underwriting issue if the number of vehicles is greater than 5. If the number of vehicles is 6, then the default approval amount is 7.

Note: If the issue already has an approval, the default logic does not apply. PolicyCenter presents parameters consistent with the previous approval.

See also

- “Configuring Underwriting Issue History” on page 465 in the *Configuration Guide*
- “Underwriting Issue Type System Table” on page 467 in the *Configuration Guide*
- “Authority Profiles” on page 655 for more information about authority profiles in the PolicyCenter application and how to work with them.
- “Configuring Authority Grants” on page 456 in the *Configuration Guide*

Working with Underwriting Referral Reasons

Although PolicyCenter raises issues automatically, you can also raise issues manually. Underwriting referral reasons cause underwriting issues to be raised the next time a policy transaction is run on for that policy.

1. Go to an account and select a policy.
2. Click Risk Analysis in the left sidebar. The UW Referral Reasons tab displays all underwriting referral reasons on the policy.
3. Click the Add UW Referral Reason button.

4. Enter an issue type, and short and long description.

You can close a underwriting referral reason, open a closed one, or view details.

Quoting and Rating

When you obtain a quote for a policy, PolicyCenter rates the policy to determine the cost. PolicyCenter provides a default rating system for demonstration purposes which provides rating for all lines of business in the default configuration. You can customize the default rating system or develop your own rating system.

Guidewire Rating Management also provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter. For more information, see “Rating Management” on page 529.

This topic includes:

- “Working with Quotes” on page 419
- “Entities Associated with Costs and Transactions” on page 420
- “Calculating Transactions” on page 434
- “Internal Tools for Rating: Financial Transactions Screen” on page 439

See also

- “Rating Integration” on page 407 in the *Integration Guide* for information on the rating plugins and how to integrate your own rating engine with PolicyCenter.
- “Rating Overrides” on page 445
- “Rating Management” on page 529

Working with Quotes

This topic describes how to work with the quote screen.

Working with the Quote Screen

The **Quote** screen displays total premium, taxes and surcharges, and total cost for a policy period. PolicyCenter displays details of the quote in the **Policy Premium** tab at the bottom of the screen. The **Quote** screen varies based on the line of business. For example, the **Quote** screen for a personal auto policy displays coverages by vehicle and garage location in the **Policy Premium** tab.

Entities Associated with Costs and Transactions

The PolicyCenter financial system is responsible for obtaining the costs for a policy and determining the transactions necessary to achieve those costs. Costs are part of the policy. A foreign key links a cost to the policy element for which the cost provides a price. Each line of business implements its own costs with a subtype hierarchy that fits with its policy model.

Guidewire defines costs and transactions as follows:

Cost	A <i>cost</i> represents a unit of price for a specific combination of policy elements for a specific period of effective time. A cost is a discrete unit which cannot be broken up into smaller units. The rating system plugin in PolicyCenter or an external rating system (in production environments) returns the costs including the effective periods and any prorated amounts. Costs attach directly to things that have a price, such as a PersonalVehicle or a PersonalAutoCov or a join between the two. There is a separate cost table for each line of business and there are subtypes of the Cost entity for each type of cost.
Transaction	A <i>transaction</i> represents a line item in a running log of pricing changes. You can retrieve transactions from the policy period, and transactions point to the costs that they offset or onset. <i>Onset</i> transactions point to costs in the same period. <i>Offset</i> transactions point to a cost in the based-on period.

The following are examples of how PolicyCenter handles costs and transactions.

- The policy period can be shortened by a cancellation or a policy change made part way through the period. If a cost is reduced because of a shortened policy period, then a new transaction partially offsets the original cost. For example, a coverage originally costs \$100 for a one year policy period. The policy is canceled six months into the policy period. PolicyCenter creates an offset transaction for -\$50.
- The cost changes because there is a new price. For example, a coverage originally costs \$100. A policy change increases the coverage for the whole policy period, resulting in a higher price of \$110. PolicyCenter creates an offset transaction for the prior cost (-\$100) and a new onset transaction (+\$110) for the new cost.

In the PolicyCenter default configuration, costs and transactions are implemented as *delegates*. Delegates are special virtual entities that define key properties or methods for a generic type. You cannot use delegates directly, but you can create an *owning class* that implements the delegate. Each line of business contains its own tables of costs and transactions which are implemented as owning classes to the Cost and Transaction delegates.

Note: The following sections provide information on multiple lines of business. The variable *LOB* stands in for the line (such as PA, BOP, or WC) in file and path names.

This topic contains the following:

- “Cost Delegate” on page 420
- “Transaction Delegate” on page 422
- “Policy Period Fields for Costs and Transactions” on page 423
- “Cost and Transaction Model for Businessowners Line” on page 424
- “Cost and Transaction Model for Commercial Property Line” on page 426
- “Cost and Transaction Model for General Liability Line” on page 427
- “Cost and Transaction Model for Inland Marine Line” on page 430
- “Cost and Transaction Model for Personal Auto Line” on page 431
- “Cost and Transaction Model for Workers’ Compensation Line” on page 433

Cost Delegate

Costs are created when the policy is rated.

The **Cost** delegate is the basic building block for a cost. The delegate provides the common financial columns and behaviors. The delegate assumes that the implementing line decides how the line relates to the building, vehicle, coverage or other item that is being priced.

The **Cost** delegate:

- Has a property that indicates whether the cost is prorated or flat. If the cost is prorated, there is a property for the proration factor.
- Has an effective and expiration date.
- Can determine if it is fundamentally the same as another cost through the **CostKey** property.
- Can create onset or offset transactions for particular subperiods within its effective period. The transactions created are defined by its **LOBCostAdapter**. You can view this in Studio by going to **Classes** → **gw** → **lob** → **LOB** → **financials**.
- Can calculate the prorated amount from term amount and effective date.
- Provides additional Gosu functionality defined by the **LOBCostMethods** Gosu interface. You can find this interface in Guidewire Studio by going to **configuration** → **gsrc** and opening this file in the **gw.lob.LOB.financials** package. The user interface is the primary user of this interface. For example, the interface provides properties that can filter costs in the user interface.

In the base configuration, each line has one abstract supertype table (**LOBCost**) that defines all the costs for that line and contains the following key properties:

Property	Description
Basis	The basis for the cost over the rated term. The basis type itself may vary.
ActualBaseRate	The base rate, before applying modifier factors, for the cost over the rated term.
ActualAdjRate	The adjusted rate, after applying modifier factors, for the cost over the rated term.
ActualTermAmount	The cost over a rated term. If the cost is prorated, the unprorated amount.
EffectiveDate	The date this cost becomes effective.
ExpirationDate	The date this cost expires.
NumDaysInRatedTerm	The number of days in the standard term used for determining the term amount.
ActualAmount	The current amount of money for the effective period. If the cost is prorated, the prorated amount.
RateAmountType	Tax/surcharge, a standard premium, or a non-standard premium.

Note: An owning class for the **Cost** delegate must be an **EffDatedBean**. To view or edit the data definitions for the costs in Studio, see **LOBCost.eti** in **configuration** → **config** → **Metadata** → **Entity**.

Guidewire provides the owning classes for the **Cost** delegate as an example, based on the way policies are commonly priced for each line of business. You can model the costs to fit your business needs.

Prorated or Flat Costs

In the default configuration, a cost can be either prorated or flat.

The value of a prorated cost is calculated based on the number of days it exists on the policy. The cost is prorated by dividing the number of days in the rated policy period by the number of days in the rated policy term. For example, the cost of a coverage for the policy term is \$100. The policy term has two policy periods because of a midterm policy change effective halfway through the policy term. The number of days in both policy periods is equal to half the term. Therefore, the cost is prorated to half of \$100 or \$50 for each policy period.

The value of a flat cost is a set amount that either exists on the policy period or does not independent of the length of the policy term. The cost is always the same amount regardless of when the cost appears on the policy. An example of a flat cost might be a fee which exists whenever the insured adds an additional insured to the

policy. PolicyCenter does not adjust the amount of a flat cost based the length of the effective policy period. In other words, PolicyCenter does not prorate the cost.

Some qualities of flat costs include:

- If added midterm, the full amount of the flat cost is charged.
- If removed midterm, the full amount of the flat cost is still charged unless the cost is removed on the same effective date that it was added.
- If you lengthen or shorten the policy term, the amount of the flat cost does not change.
- In a pro rata cancellation, a flat cost remains at the full amount. However, PolicyCenter reverses the flat cost if the following occurs:
 - A policy change adds a flat cost effective after submission.
 - A pro rata cancellation occurs effective on or before the policy change effective date.
- Upon flat cancellation, refund a flat cost at the full amount.
- A policy change that affects a flat cost typically generates a pair of transactions. Suppose a policy has a flat-rated coverage, and you make a policy change which affects the cost for that coverage. PolicyCenter generates a pair of transactions related to that coverage. The first transaction reverses the cost for the old date range. The second transaction reapplies the cost to the new date range.
- A flat cost is charged twice if:
 - You put a flat cost on the policy.
 - Remove it at a later effective date.
 - Add it again at another later effective date.

On the `Cost` object, the `ProrationMethod` property specifies whether the cost is prorated or flat. The property contains a `ProrationMethod` typekey. In the default configuration, this typekey can be one of the following values: `ProRataByDays` (a prorated cost) and `Flat` (a flat cost).

Rating Systems and Flat Costs

With Guidewire Rating Management, you can define and rate flat costs. The personal auto line of business contains an example of a flat cost in the **Mexico Coverage - Limited** coverage. In the policy line in Studio, this coverage is defined just like any other coverage. The coverage does not have a flat-rated field. You specify the flat cost in the rate routine for this coverage. For more information, see “Specifying a Flat-rated Coverage in a Rate Routine” on page 576.

If you do not use Guidewire Rating Management, you can define flat costs, but you must configure the rating for those flat costs. You can define costs as flat costs in Gosu code by setting the `ProrationMethod` property to `Flat` on the cost data object. The system table rating plugin implementation contains code to handle flat costs in the `CostData` class. The system table rating plugin does not contain any examples of flat costs. Although the personal auto line of business includes the **Mexico Coverage - Limited** coverage, you must configure the system table rating plugin to rate it as a flat cost.

Cost Adapter

The `Cost` delegate requires that the owning class provide an implementation of the following interface:

```
gw.api.domain.financials.CostAdapter
```

This interface defines the services that the delegate needs to work properly. In the base configuration, PolicyCenter class `LOBCostAdapter` implements this interface. You can view the cost adapter in Studio by going to `configuration` → `gsrc` and opening this file in the `gw.lob.LOB.financials` package.

Transaction Delegate

Transactions are created after the policy is rated.

The Transaction delegate is the basic building block for a transaction. The delegate provides the common financial columns and behaviors, and the implementing line decides how to hook into the policy graph. The Transaction for each line of business needs a foreign key to the Cost for that line of business. The Cost points to other tables for that line of business.

The Transaction delegate:

- Knows if it is a prorated section of the cost it modifies, and if so, what the proration factor is.
- Can retrieve the cost it modifies.

In the base configuration, each line has one transaction table (*LOBTransaction*) that has a non-effdated foreign key to the cost for that line and contains the following key properties:

Property	Description
Amount	The transaction amount for the effective time EffDate, ExpDate.
EffDate	The date on which the transaction becomes effective.
ExpDate	The date on which the transaction expires.

In general, you do not need to modify the owning class to the Transaction delegate in your custom configuration.

Transaction Adapter

The Transaction delegate requires that the owning class provide an implementation of the following interface.
`gw.api.domain.financials.TransactionAdapter`

This interface defines the services that the delegate needs. In the base configuration, PolicyCenter class *LOBTransactionAdapter* implements this interface.

Policy Period Fields for Costs and Transactions

The PolicyPeriod entity has several fields which contain the total value of costs and transactions. These fields appear on the Quote screen.

After rating, PolicyCenter calculates these total value fields by adding up the underlying costs or transactions. PolicyCenter stores these calculated total value fields for better performance. For example, you can use these fields to display the total change in cost for a policy change. You can also use these fields to display the total premium for a submission. If you use the total value fields, PolicyCenter does not need to recalculate the amount.

Two cost fields store a total for the costs and represent the full price for the entire period. These values appear on the Quote screen in the Total Premium and Total Cost fields and in the Policy Premium tab. The fields for costs are:

- `TotalCostRPT` – Total value of all costs, including taxes and fees.
- `TotalPremiumRPT` – Total value of all premium costs.

Two transaction fields are calculated from the transactions and store the change in transaction cost for the policy transaction. The transaction cost appears on the Quote screen in the Change in Cost field. Both fields appear in Quote screen on the Cost Change Detail tab for mid-term policy transactions such as a policy change. The fields for transactions are:

- `TransactionCostRPT` – Total value of all transactions, including taxes and fees.
- `TransactionPremiumRPT` – Total value of all premium transactions.

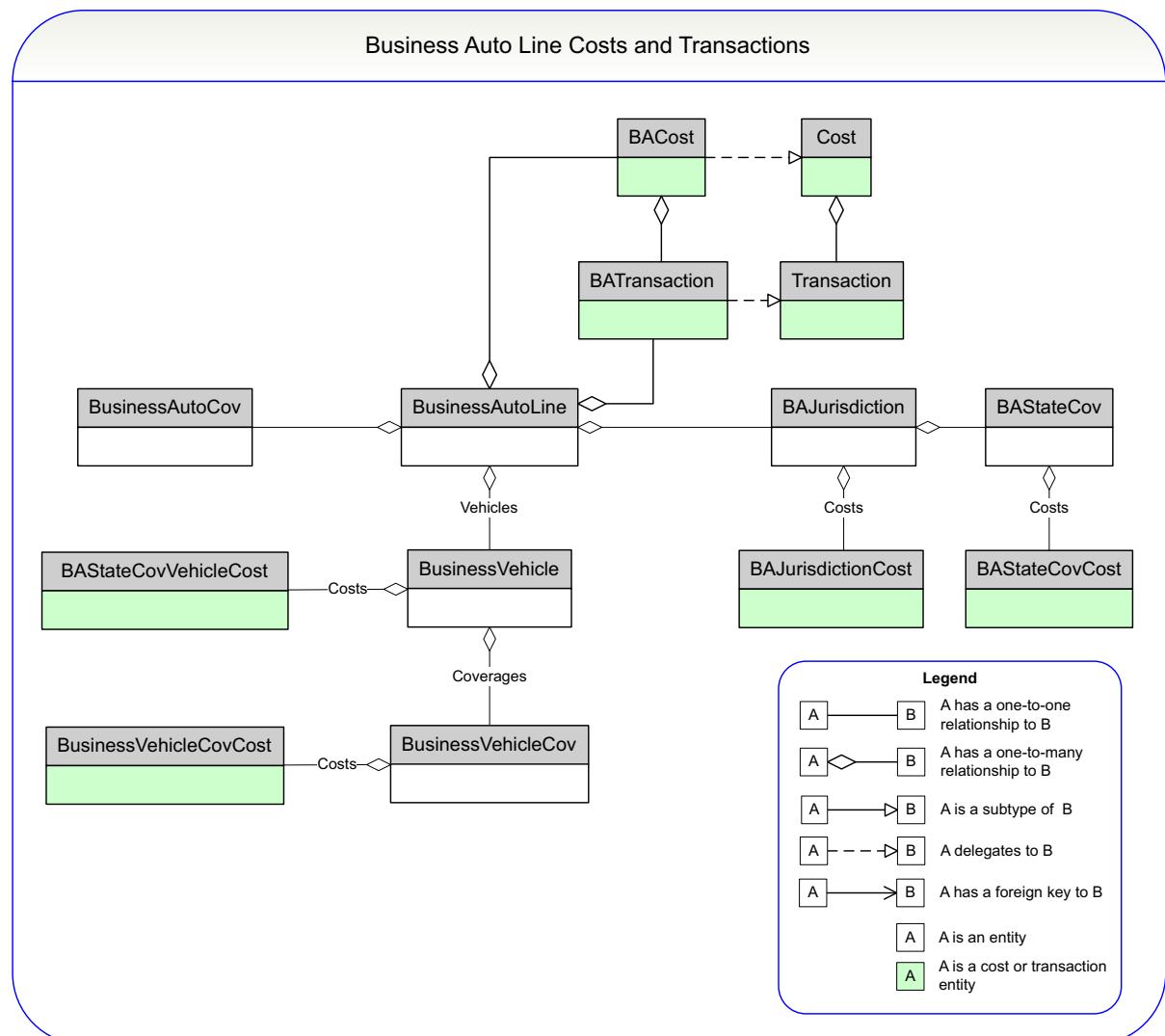
PolicyCenter calculates these fields when the policy is quoted. The calculation occurs after rating the policy and if there is a valid quote. The `denormalizeFinancialTotals` method calculates these field values. See `gw.job.QuoteProcess.gs` located in configuration → gsrc in Studio.

Cost and Transaction Model for Business Auto Line

The following diagram illustrates how the various types of costs and transactions interact in the business auto line. The **BusinessAutoLine** is a subtype of **PolicyLine**. As you can see in the diagram, **BusinessAutoLine** connects directly to financial array **BACost**. The **BACost** entity has an array key to **BATransaction**.

The **BusinessAutoLine** entity has a derived array to **BATransaction**. The **BusinessAutoLine** gets its derived array of transactions by asking the **PolicyPeriod** for all of its **BATransaction** entities. The **PolicyPeriod** entity has an array key to **BATransaction**.

The line of business has a derived array of the transactions to support mid-term removal of a line in a multi-line policy. In a multi-line policy, you can delete a line entirely. If you do that in a mid-term policy change, you may have one or more offset transactions to return premium that is no longer justified for the line. These offset transactions cannot be stored on the non-existent line. Therefore, the offset transactions are stored on the policy period.

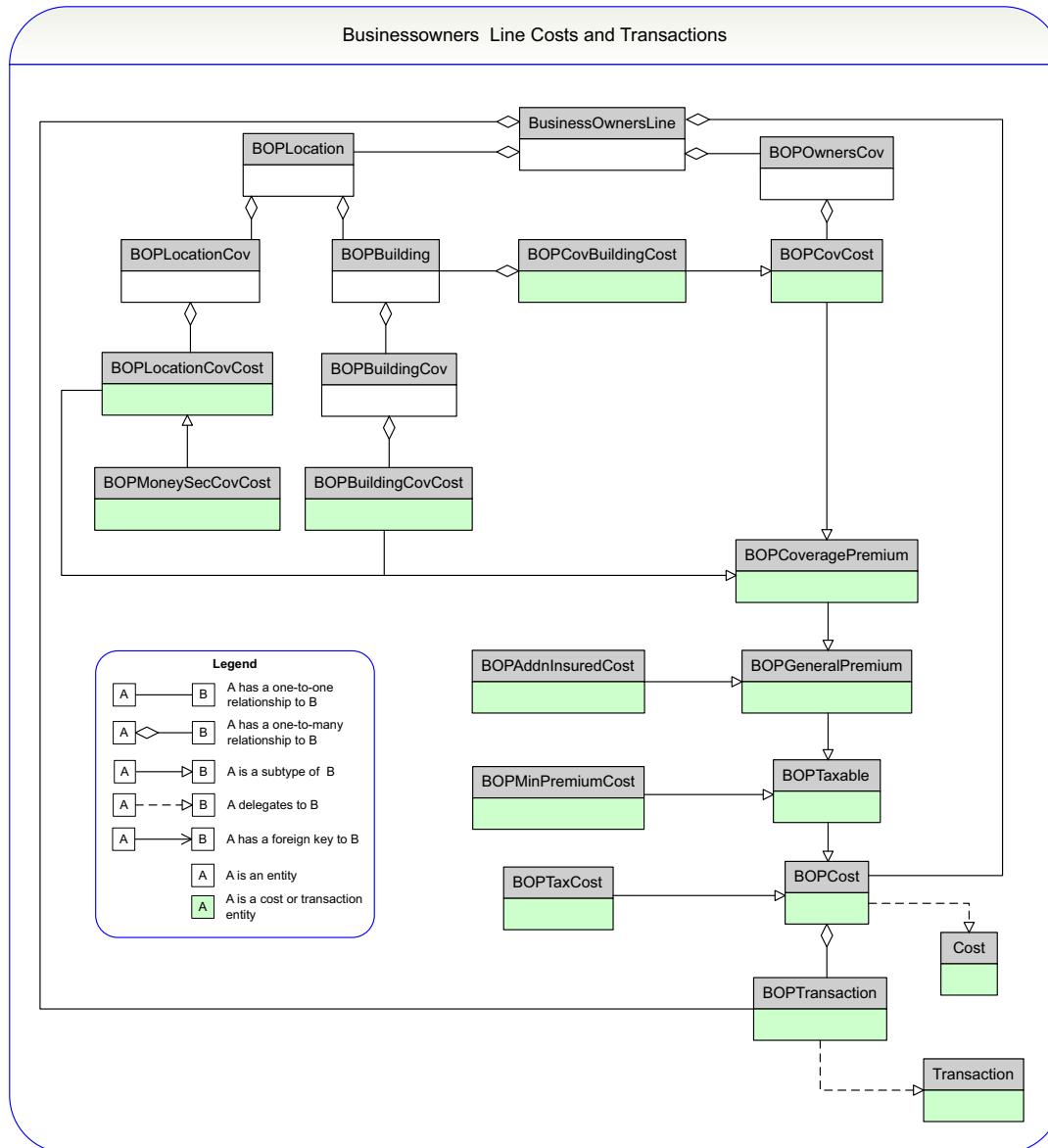


Cost and Transaction Model for Businessowners Line

The following diagram illustrates how the various types of costs and transactions interact in the businessowners line. The **BusinessOwnersLine** is a subtype of **PolicyLine**. As you can see in the diagram, **BusinessOwnersLine** connects to the financial array **BOPCost**. The **BOPCost** entity has an array key to **BOPTransaction**.

The `BusinessOwnersLine` entity has a derived array to `BOPTransaction`. The `BusinessOwnersLine` gets its derived array of transactions by asking the `PolicyPeriod` for all of its `BOPTransaction` entities. The `PolicyPeriod` entity has an array key to `BOPTransaction`.

The line of business has a derived array of the transactions to support mid-term removal of a line in a multi-line policy. In a multi-line policy, you can delete a line entirely. If you do that in a mid-term policy change, you may have one or more offset transactions to return premium that is no longer justified for the line. These offset transactions cannot be stored on the non-existent line. Therefore, the offset transactions are stored on the policy period.



Subtypes

The concrete `Cost` subtypes are:

- `BOPLocationCovCost`
- `BOPMoneySecCovCost`
- `BOPBuildingCovCost`

- BOPCovBuildingCost
- BOPCovCost
- BOPAddn1InsuredCost
- BOPMinPremiumCost
- BOPTaxCost

The abstract Cost subtypes are:

- BOPCoveragePremium
- BOPGeneralPremium
- BOPTaxable
- BOPCost

The abstract cost subtypes extend one another in the hierarchy. This hierarchy makes it easy to get all costs of at different levels of the hierarchy. For example, if you get all BOPTaxable costs, you also get all BOPCoveragePremium and BOPGeneralPremium costs. The cost supertype, BOPCost, includes costs of all abstract subtypes.

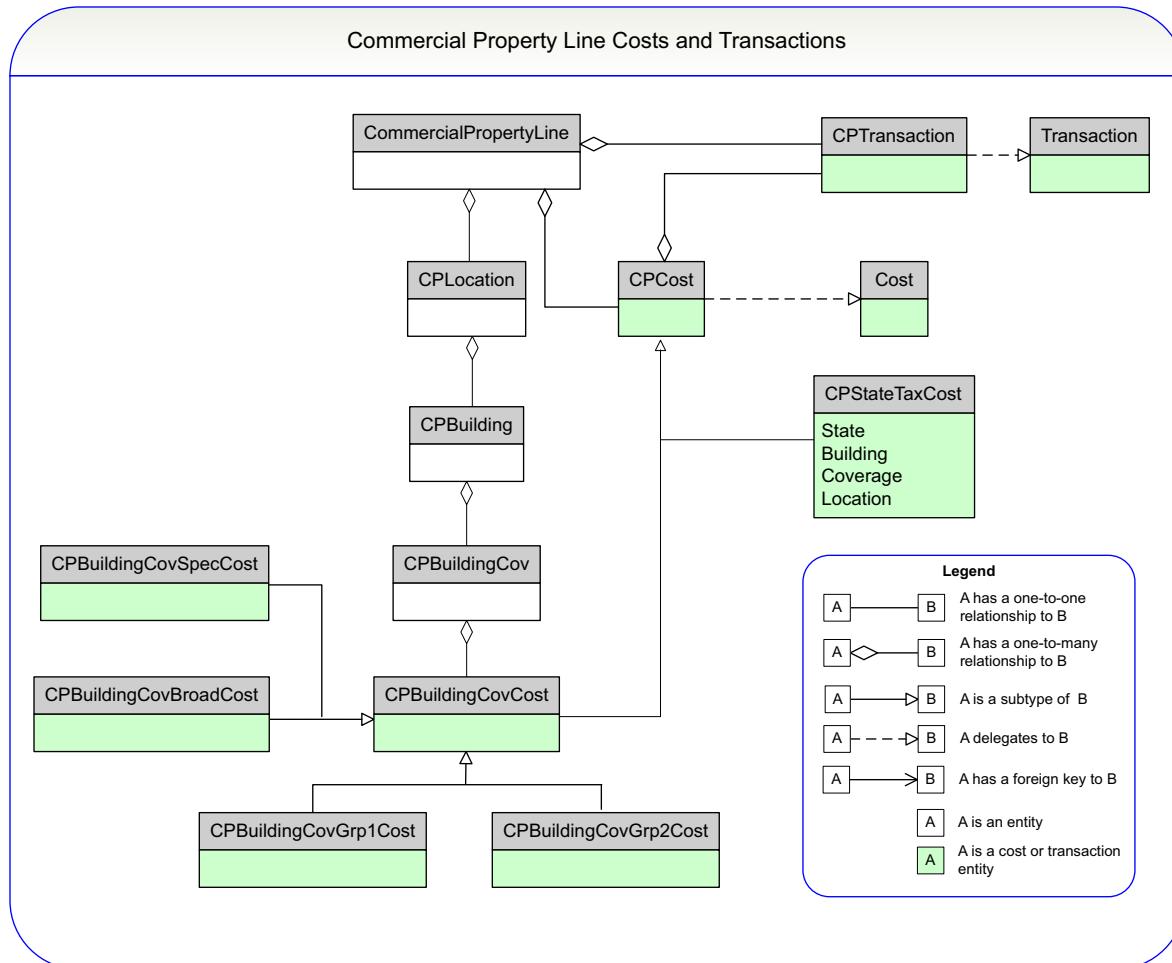
Cost and Transaction Model for Commercial Property Line

The following diagram illustrates how the various types of costs and transactions interact in the commercial property line. The `CommercialPropertyLine` is a subtype of `PolicyLine`. As you can see in the diagram, `CommercialPropertyLine` connects directly to financial array `CPCost`. The `CPCost` entity has an array key to `CPTtransaction`.

The `CommercialPropertyLine` entity has a derived array to `CPTtransaction`. The `CommercialPropertyLine` gets its derived array of transactions by asking the `PolicyPeriod` for all of its `CPTtransaction` entities. The `PolicyPeriod` entity has an array key to `CPTtransaction`.

The line of business has a derived array of the transactions to support mid-term removal of a line in a multi-line policy. In a multi-line policy, you can delete a line entirely. If you do that in a mid-term policy change, you may have one or more offset transactions to return premium that is no longer justified for the line. These offset trans-

actions cannot be stored on the non-existent line. Therefore, the offset transactions are stored on the policy period.



CPCost

The **CPCost** entity defines a cost associated with the commercial property line. The **CommercialPropertyLine** entity has an array of **CPCost** entities. Each **CPCost** entity has an array of **CPTTransaction** entities.

CPBuildingCovCost

The **CPBuildingCovCost** entity is a subtype of the **CPCost** entity, for capturing CP building coverage costs. Building costs are calculated in four separate rates, modeled as subtypes: **CPBuildingCovGrp1Cost**, **CPBuildingCovGrp2Cost**, **CPBuildingCovBroadCost**, and **CPBuildingCovSpecCost**.

CPStateTaxCost

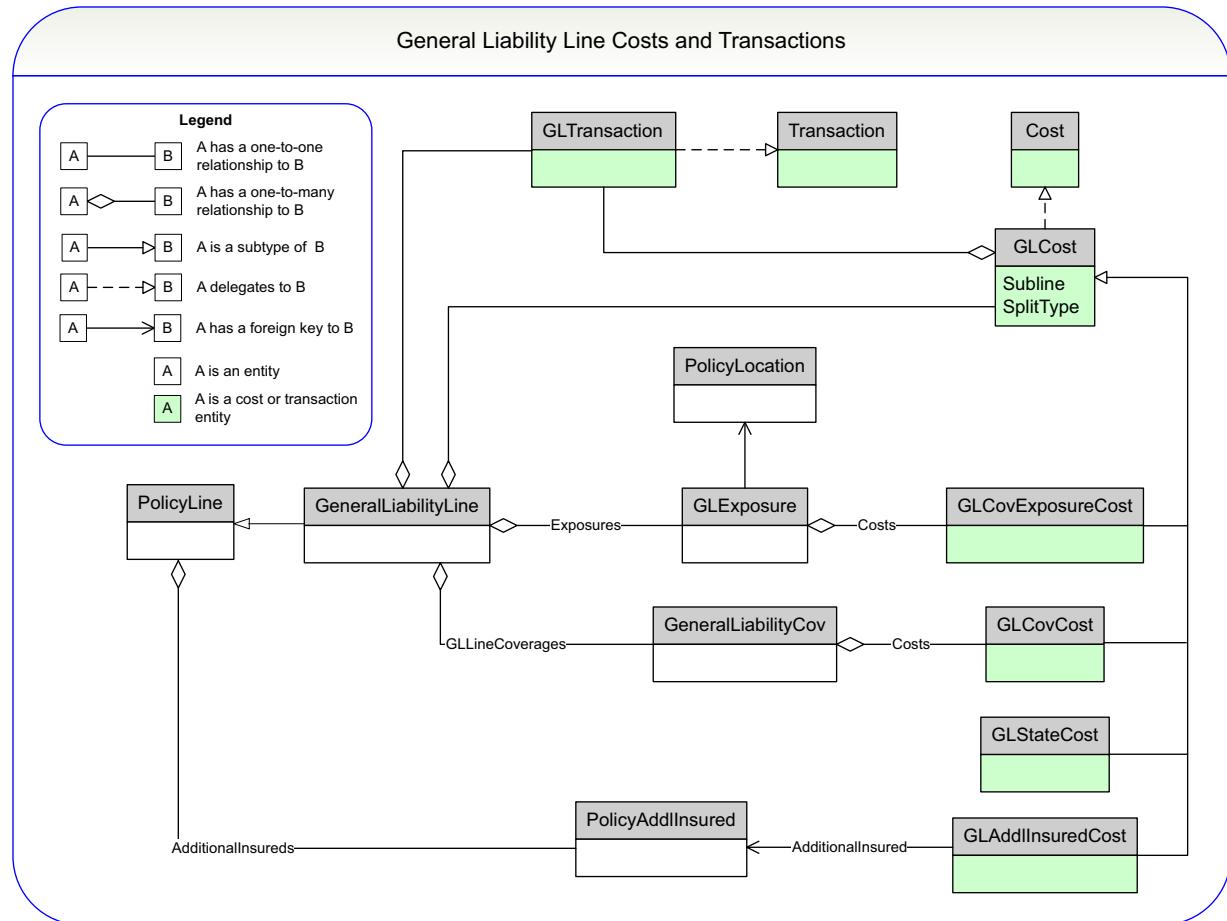
Captures the tax cost on the line per jurisdiction.

Cost and Transaction Model for General Liability Line

The following diagram illustrates how the various types of costs and transactions interact in the general liability line. The **GeneralLiabilityLine** is a subtype of **PolicyLine**. As you can see in the diagram, **GeneralLiabilityLine** connects directly to financial array **GLCost**. The **GLCost** entity has an array key to **GLTransaction**.

The GeneralLiabilityLine entity has a derived array to GLTransaction. The GeneralLiabilityLine gets its derived array of transactions by asking the PolicyPeriod for all of its GLTransaction entities. The PolicyPeriod entity has an array key to GLTransaction.

The line of business has a derived array of the transactions to support mid-term removal of a line in a multi-line policy. In a multi-line policy, you can delete a line entirely. If you do that in a mid-term policy change, you may have one or more offset transactions to return premium that is no longer justified for the line. These offset transactions cannot be stored on the non-existent line. Therefore, the offset transactions are stored on the policy period.



Sublines on Costs

Rating for a general liability exposure is often based on two rates: one rate for premises and operations and another rate for products and completed operations.

For each GLExposure and the GeneralLiabilityLine, the rating engine generates two cost rows, one for premises and operations and the other for products and completed operations. The Subline field allows you to distinguish between these rows. You can add additional typecodes to the GLCostSubline typelist, if necessary.

Values for Subline are:

- Premises – Premises and operations
- Products – Products and completed operations

Costs for Split Bodily Injury and Physical Damage Limits

The user can choose to split bodily injury and physical damage limits for general liability coverages. If the limits are split, the rating engine must calculate separate costs for each exposure. The rating engine must also set the **LiabilityLimitSplitType** field on GLCost to:

- BI – for Bodily Injury
- PD – for Property Damage
- CSL – for a Combined Single Limit

See also

- “Standard Coverages” on page 235 in the *Application Guide* for more information on the **Split BI / PD Limits** field.

Cost Examples for General Liability

Depending upon choices made by the user, the rating engine creates two or four costs per exposure.

If the user selects not to split bodily injury and physical damage limits, the rating engine creates two costs for each exposure.

On the **Coverages → Standard Coverages** tab, the

user set **Split BI / PD Limits** to **No**. On the **Exposures** screen, the user added an exposure for antique stores. The rating engine creates the following two costs for this exposure:

- Combined single limit for the premises subline with rate 0.984
- Combined single limit for the product subline with rate 0.656

Quote																																																										
Submission Number		3940582		Total Premium		\$164.00																																																				
Policy Period		03/03/2010 - 03/03/2011		Taxes & Surcharges		\$12.00																																																				
Policy Premium																																																										
California																																																										
Exposure Premium																																																										
<table border="1"> <thead> <tr> <th>Loc.</th> <th>Code</th> <th>Description</th> <th>Subline</th> <th>Split</th> <th>Basis</th> <th>Rate</th> <th>Term Premium</th> <th>Eff Date</th> <th>Exp Date</th> <th>Proration</th> <th>Amount</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0004</td> <td>Antique Stores</td> <td>Premises</td> <td>CSL</td> <td>100000</td> <td>0.984</td> <td>\$98.00</td> <td>03/03/2010</td> <td>03/03/2011</td> <td>1.0000</td> <td>\$98.00</td> </tr> <tr> <td>1</td> <td>0004</td> <td>Antique Stores</td> <td>Products</td> <td>CSL</td> <td>100000</td> <td>0.656</td> <td>\$66.00</td> <td>03/03/2010</td> <td>03/03/2011</td> <td>1.0000</td> <td>\$66.00</td> </tr> <tr> <td colspan="3">Subtotal</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>\$164.00</td></tr> </tbody> </table>											Loc.	Code	Description	Subline	Split	Basis	Rate	Term Premium	Eff Date	Exp Date	Proration	Amount	1	0004	Antique Stores	Premises	CSL	100000	0.984	\$98.00	03/03/2010	03/03/2011	1.0000	\$98.00	1	0004	Antique Stores	Products	CSL	100000	0.656	\$66.00	03/03/2010	03/03/2011	1.0000	\$66.00	Subtotal											\$164.00
Loc.	Code	Description	Subline	Split	Basis	Rate	Term Premium	Eff Date	Exp Date	Proration	Amount																																															
1	0004	Antique Stores	Premises	CSL	100000	0.984	\$98.00	03/03/2010	03/03/2011	1.0000	\$98.00																																															
1	0004	Antique Stores	Products	CSL	100000	0.656	\$66.00	03/03/2010	03/03/2011	1.0000	\$66.00																																															
Subtotal											\$164.00																																															
Other Premium and Surcharges																																																										

If the user chooses to split bodily injury and physical damage limits, the rating engine creates four costs for each exposure.

On the **Coverages → Standard Coverages** tab, the user set **Split BI / PD Limits** to **Yes**. On the **Exposures** screen, the user added an exposure for antique stores. The rating engine creates the following four costs for this exposure:

- Bodily injury for the premises subline with rate 0.7895
- Physical damage for the premises subline with rate 0.2045
- Bodily injury for the product subline with rate 0.5263

- Physical damage for the product subline with rate 0.1364

Loc.	Code	Description	Subline	Split	Basis	Rate	Term Premium	Eff Date	Exp
1	0004	Antique Stores	Premi...	BL	20000	0.7895	\$16.00	08/13/20...	08/13/2014
1	0004	Antique Stores	Premi...	PD	20000	0.2045	\$4.00	08/13/20...	08/13/2014
1	0004	Antique Stores	Produ...	BL	20000	0.5263	\$11.00	08/13/20...	08/13/2014
1	0004	Antique Stores	Produ...	PD	20000	0.1364	\$3.00	08/13/20...	08/13/2014
		Subtotal							

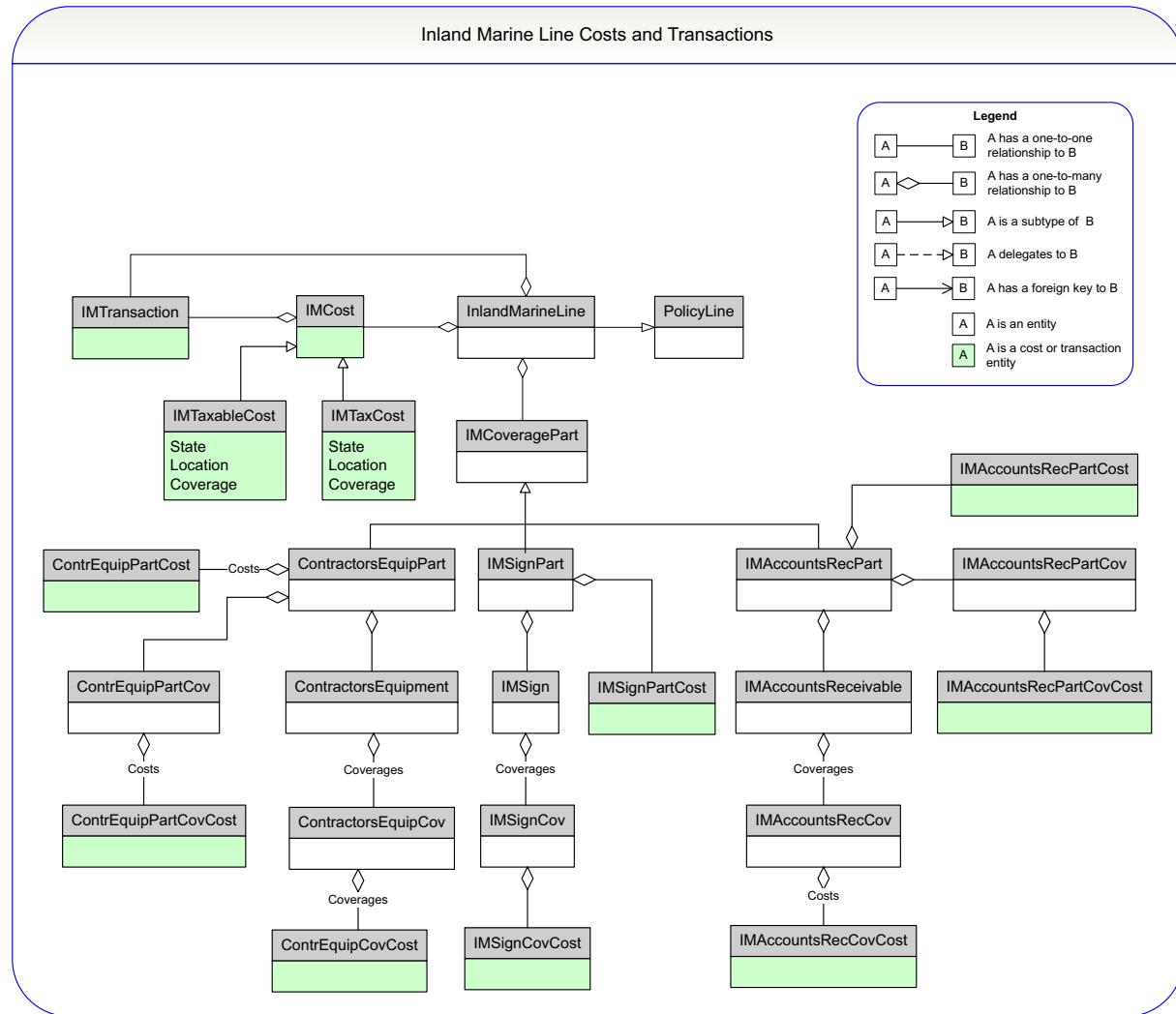
Cost and Transaction Model for Inland Marine Line

The following diagram illustrates how the various types of costs and transactions interact in the inland marine line. The `InlandMarineLine` is a subtype of `PolicyLine`. As you can see in the diagram, `InlandMarineLine` entity has an array key to the financial array `IMCost`. The `IMCost` entity has an array key to `IMTransaction`.

The `InlandMarineLine` entity has a derived array to `IMTransaction`. The `InlandMarineLine` gets its derived array of transactions by asking the `PolicyPeriod` for all of its `IMTransaction` entities. The `PolicyPeriod` entity has an array key to `IMTransaction`.

The line of business has a derived array of the transactions to support mid-term removal of a line in a multi-line policy. In a multi-line policy, you can delete a line entirely. If you do that in a mid-term policy change, you may have one or more offset transactions to return premium that is no longer justified for the line. These offset trans-

actions cannot be stored on the non-existent line. Therefore, the offset transactions are stored on the policy period.



Cost and Transaction Model for Personal Auto Line

The following diagram illustrates how the various types of costs and transactions interact in the personal auto line. The **PersonalAutoLine** is a subtype of **PolicyLine**. As you can see in the diagram, **PersonalAutoLine** connects directly to financial arrayPACost. The **PACost** entity has an array key to **PATransaction**.

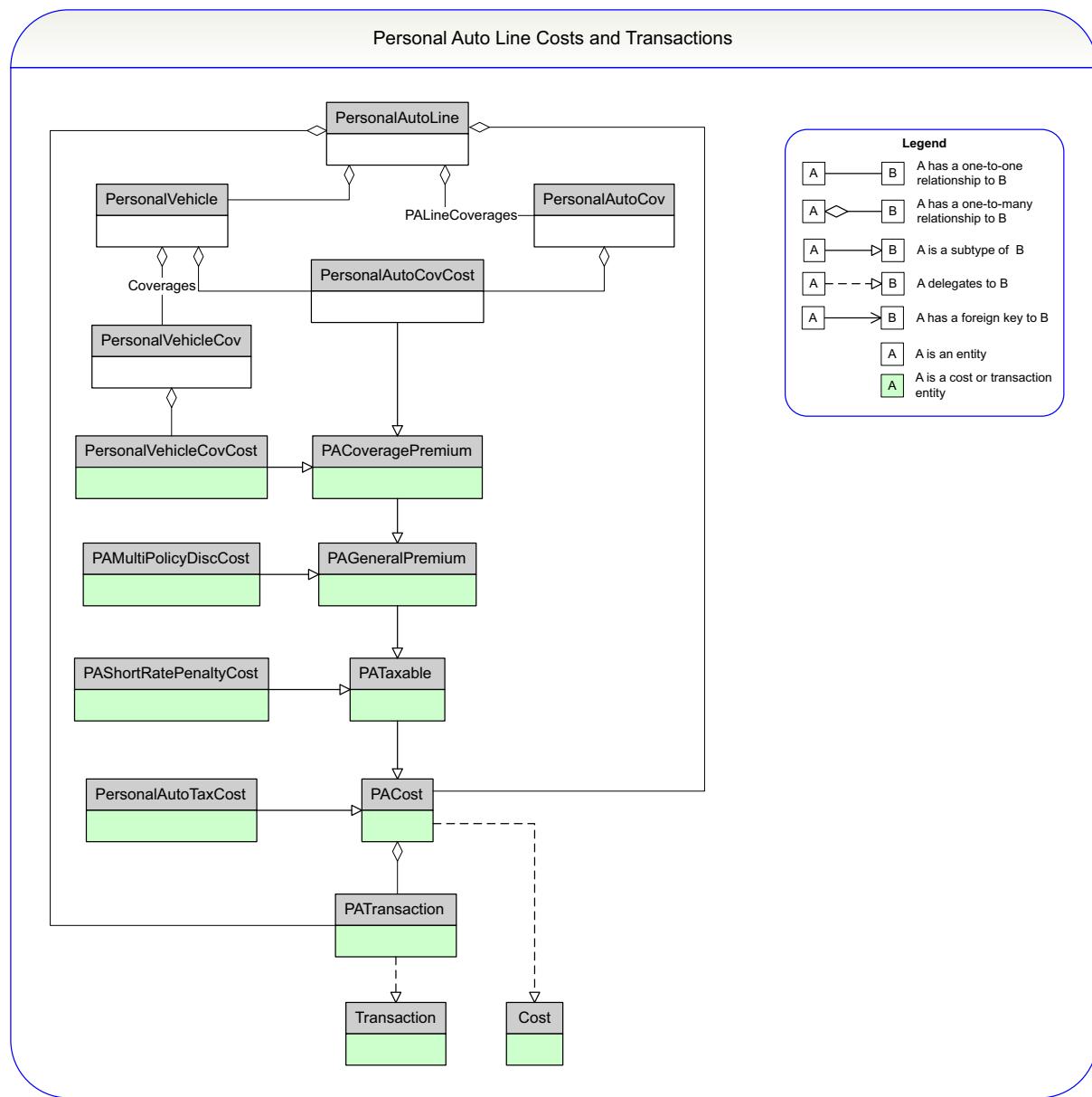
The **PersonalAutoLine** entity has a derived array to **PATransaction**. The **PersonalAutoLine** gets its derived array of transactions by asking the **PolicyPeriod** for all of its **PATransaction** entities. The **PolicyPeriod** entity has an array key to **PATransaction**.

The line of business has a derived array of the transactions to support mid-term removal of a line in a multi-line policy. In a multi-line policy, you can delete a line entirely. If you do that in a mid-term policy change, you may have one or more offset transactions to return premium that is no longer justified for the line. These offset transactions cannot be stored on the non-existent line. Therefore, the offset transactions are stored on the policy period.

Although not shown on the diagram:

- **PersonalAutoCov** delegates to **Coverage**.

- PersonalVehicle delegates to Coverable.



Subtypes

The concrete Cost subtypes are:

- PAMultiPolicyDiscCost
- PAShortRatePenaltyCost
- PersonalAutoCovCost
- PersonalAutoTaxCost
- PersonalVehicleCovCost

The abstract Cost subtypes are:

- PACoveragePremium
- PAGeneralPremium
- PATaxable
- PACost

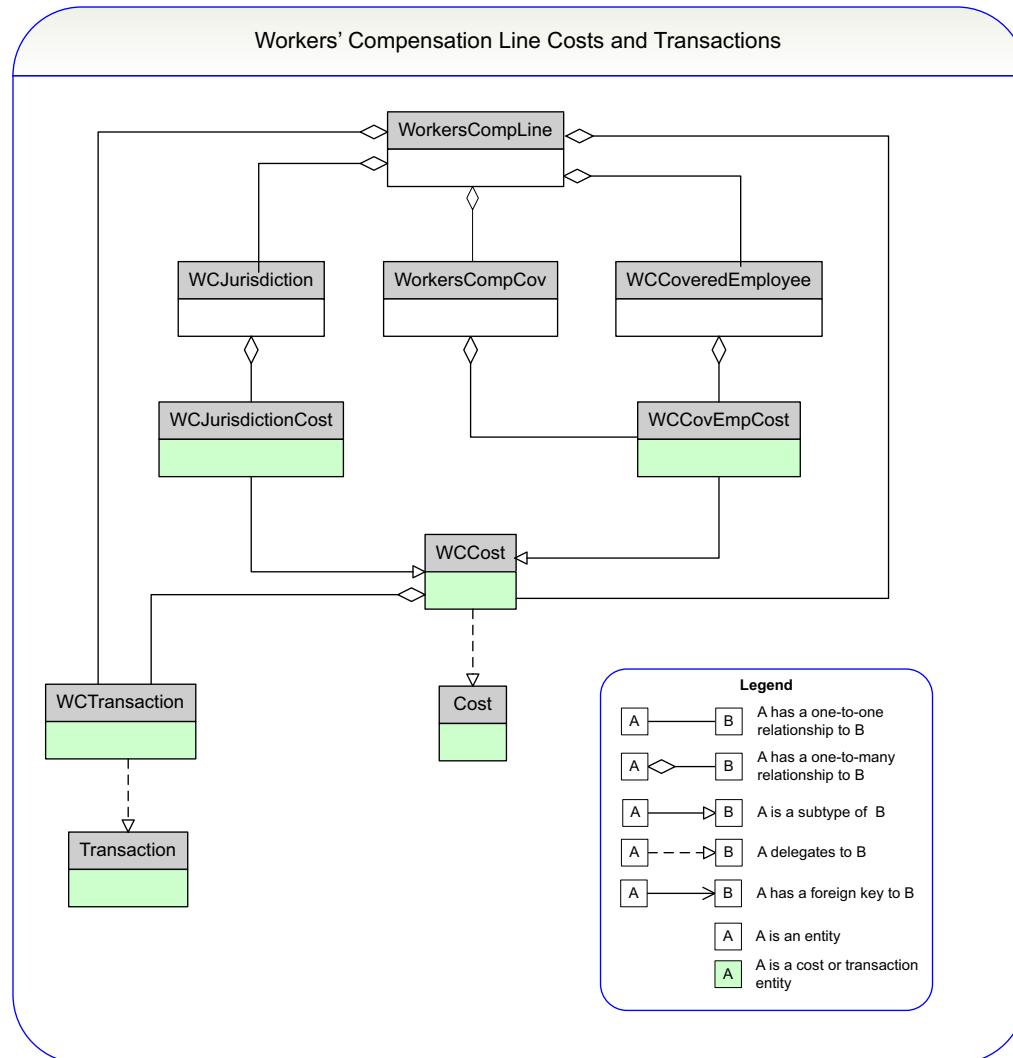
The abstract cost subtypes extend one another in the hierarchy. This hierarchy makes it easy to get all costs of at different levels of the hierarchy. For example, if you get all PATaxable costs, you also get all PACoveragePremium and PAGeneralPremium costs. The cost supertype, PACost, includes costs of all abstract subtypes.

Cost and Transaction Model for Workers' Compensation Line

The following diagram illustrates how the various types of costs and transactions interact in the workers' compensation line. The **WorkersLine** is a subtype of **PolicyLine**. As you can see in the diagram, **WorkersCompLine** connects to financial array **WCCost**. The **WCCost** entity has an array key to **WCTransaction**.

The **WorkersCompLine** entity has a derived array to **WCTransaction**. The **WorkersCompLine** gets its derived array of transactions by asking the **PolicyPeriod** for all of its **WCTransaction** entities. The **PolicyPeriod** entity has an array key to **WCTransaction**.

The line of business has a derived array of the transactions to support mid-term removal of a line in a multi-line policy. In a multi-line policy, you can delete a line entirely. If you do that in a mid-term policy change, you may have one or more offset transactions to return premium that is no longer justified for the line. These offset transactions cannot be stored on the non-existent line. Therefore, the offset transactions are stored on the policy period.



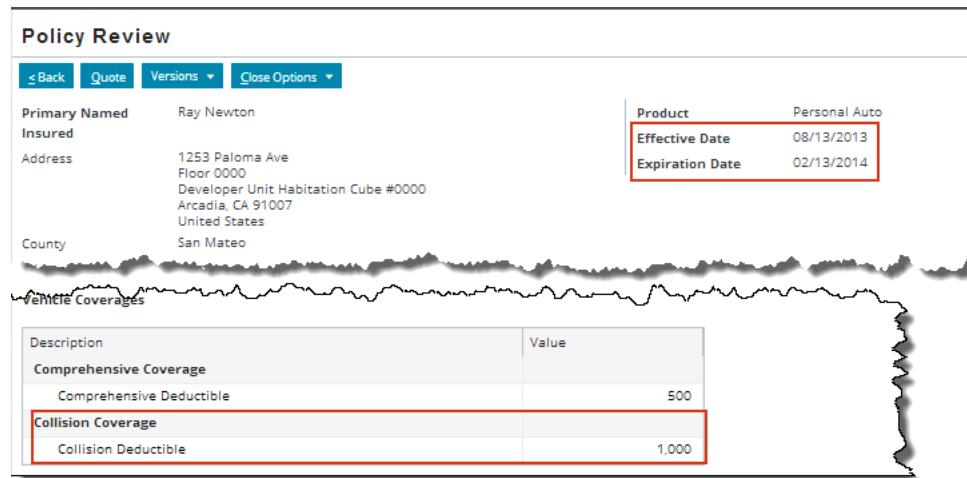
Calculating Transactions

After the rating system creates the costs, the default application constructs *transactions* for the current *policy transaction*. Transactions represent changes in cost. Policy transactions coordinate all the work associated with creating a new policy period and modifying the policy. Transactions are sent to a billing system or used them for premium accounting. The base configuration provides demonstration code for rating and an integration with Guidewire BillingCenter. For more information, see “Billing System Integration” on page 701.

This topic provides an example showing costs and transactions in a personal auto submission policy transaction followed by a policy change transaction. The policy period is six months. In the submission, the customer selects collision coverage with a \$1000 deductible. Later, the customer calls and asks to lower the deductible to \$250 at the beginning of the fourth month (halfway through the policy period). The agent submits a policy change.

Note: In PolicyCenter, the **Policy Premium** tab displays costs, and the **Cost Change Detail** tab displays transactions.

In the submission policy transaction, you create a six-month policy choosing collision coverage with a \$1000 deductible.



Vehicle Coverages	
Description	Value
Comprehensive Coverage Comprehensive Deductible	500
Collision Coverage Collision Deductible	1.000

The **Policy Premium** tab on the **Quote** screen below shows the costs for the coverages and tax highlighted in red. The cost for collision coverage is \$21 and taxes are \$49.

Quote

[Back](#) [Next >](#) [Release Lock](#) [Edit Policy Transaction](#) [Save Draft](#) [Versions](#) [Bind Options](#) [Close Options](#) [Print Quote](#)

Submission Number	634621	Total Premium	\$673.00
Policy Period	08/13/2013 - 02/13/2014	Taxes & Surcharges	\$49.00
Primary Named Insured	Ray Newton	Total Cost	\$722.00

Address: 1253 Paloma Ave, Arcadia, CA 91007

Company: AutoZone Insurance Services Inc.

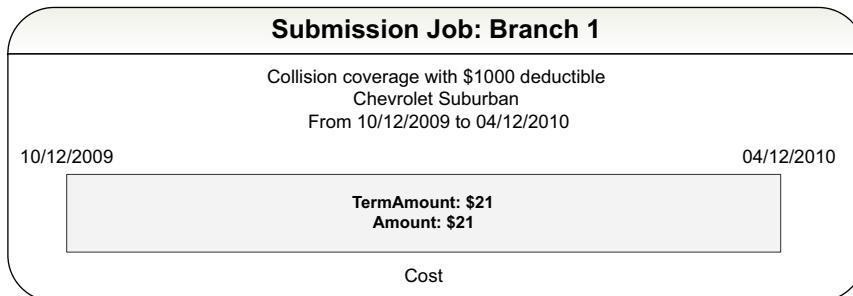
Policy Premium

Garage 1: 1253 Paloma Ave, Floor 0000, Developer Unit Habitation Cube #0000, Arcadia, CA 91007

Vehicle 1

Model Year	2003	Make	Chevrolet	Model	Suburban	VIN	12345
Description		Premium					
Liability - Bodily Injury and Property Damage Coverage		\$588.00					
Medical Payments Coverage		-					
Uninsured Motorist - Bodily Injury Coverage		\$37.00					
Uninsured Motorist - Property Damage Coverage		\$12.00					
Comprehensive Coverage		\$15.00					
Collision Coverage		\$21.00					
Subtotal		\$673.00					
Premium Subtotal	\$673.00						
CA Tax	\$49.00						

PolicyCenter creates a policy period branch with the cost for collision coverage as shown in the following diagram.



The **Summary** screen shows the sum of all transactions for the submission policy transaction.

Summary

Account Information		Period	
Account Number	C000143542	Effective Date	08/13/2013
Account Name	Ray Newton	Expiration Date	02/13/2014
Address	1253 Paloma Ave Floor 0000 Developer Unit Habitation Cube #0000 Arcadia, CA 91007 United States	Term Number	1
County	San Mateo	Total Premium	\$673.00
Address Type	Home	Taxes And Fees	\$49.00
Address Description	Created by the Address Builder with code 0	Total Cost	\$722.00
Official IDs		Earned Premium	
SSN	342-56-8729	Personal Auto Line	
		Calculate Earned Amount as of different date	
		Producer of Record	

Completed Policy Transactions

Compare	Period Eff Date	Trans Eff Date	Trans Close Date	Type	Transaction #	Premium	Comment
	08/13/2013	08/13/2013	08/13/2013	Submission	634621	\$722.00	

Pending Policy Transactions

At a later time, the insured calls to request a decrease in the collision coverage deductible from \$1000 to \$250 beginning three months from the policy effective date. The agent starts a policy change. The **Policy Premium** tab on the **Quote** screen displays two prorated costs for collision coverage. One cost is for the first three months, and the other is for second three months of the policy period.

Quote

Policy Number		Transaction Effective Date	
Policy Number	5690201834	11/13/2013	
Policy Period	08/13/2013 - 02/13/2014	Date	
Primary Named Insured	Ray Newton	Total Premium \$682.00	
Address	1253 Paloma Ave Floor 0000 Developer Unit Habitation Cube #0000 Arcadia, CA 91007 United States	Taxes & Surcharges \$49.00	
		Total Cost \$731.00	
		Change in Cost \$9.00	

Company

Policy Premium	Cost Change Detail						
Garage 1: 1253 Paloma Ave, Floor 0000, Developer Unit Habitation Cube #0000, Arcadia, CA 91007							
Vehicle 1							
Model Year	2003	Make	Chevrolet	Model	Suburban	VIN	12345
Description		Premium	Amount	Eff Date	Exp Date	Proration	
Liability - Bodily Injury and Property Damage Coverage		\$588.00	\$588.00	08/13/20...	02/13/20...	1.0000	
Medical Payments Coverage		-	-	08/13/20...	02/13/20...	1.0000	
Uninsured Motorist - Bodily Injury Coverage		\$37.00	\$37.00	08/13/20...	02/13/20...	1.0000	
Uninsured Motorist - Property Damage Coverage		\$12.00	\$12.00	08/13/20...	02/13/20...	1.0000	
Comprehensive Coverage		\$15.00	\$15.00	08/13/20...	02/13/20...	1.0000	
Collision Coverage		\$11.00	\$21.00	08/13/20...	11/13/20...	0.5000	
Collision Coverage		\$19.00	\$38.00	11/13/20...	02/13/20...	0.5000	
Subtotal		\$682.00					

PolicyCenter creates a new branch for the policy change. The new branch has two costs for collision coverage. Each cost has a prorated amount.

Policy Change Job: Branch 2		
Collision coverage deductible decreased Chevrolet Suburban \$1000 deductible from 10/12/2009 to 01/12/2010 \$250 deductible from 01/12/2010 to 04/12/2010		
10/12/2009	01/12/2010	04/12/2010
TermAmount: \$21 Amount: \$11	TermAmount: \$38 Amount: \$19	
Cost	Cost	

The database contains two costs as follows:

FixedID	ID	EffDate	ExpDate	Term amount	Amount	Car	Coverage
1	3	10/12/2000	01/12/2010	\$21	\$11	Chevrolet Suburban	Collision
1	4	01/12/2010	04/12/2010	\$38	\$19	Chevrolet Suburban	Collision

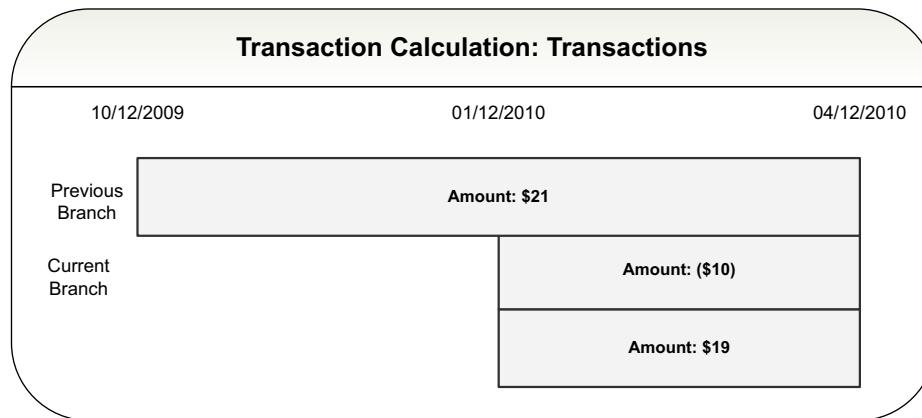
The transactions for this policy change appear on the **Cost Change Detail** tab of the **Quote** screen. The transactions are the two collision coverages and tax changes. The (\$10.00) premium is the offset transaction. The \$19.00 premium is the onset transaction.

Policy Premium		Cost Change Detail					
Vehicle 1							
Model Year	2003	Make	Chevrolet	Model	Suburban	VIN	12345
Garage Location	1253 Paloma Ave, Floor 0000, Developer Unit Habitation Cube #0000, Arcadia, CA 91007						
Description	Amount	Eff Date	Exp Date	Proration	Premium		
Collision Coverage	\$21.00	11/13/20...	02/13/20...	-0.5000	(\$10.00)		
Collision Coverage	\$38.00	11/13/20...	02/13/20...	0.5000	\$19.00		
Subtotal					\$9.00		
Premium Subtotal		\$9.00					
CA Tax		\$49.00					
CA Tax		(\$49.00)					

After the rating engine calculates the costs, PolicyCenter generates the transactions by comparing the costs in the previous and current policy period branches. In this example, the submission policy transaction created the previous, or first, branch, and the policy change created current, or second, branch.

Transaction Calculation: Costs		
10/12/2009	01/12/2010	04/12/2010
Previous Branch	TermAmount: \$21 Amount: \$21	
Current Branch	TermAmount: \$21 Amount: \$11	TermAmount: \$38 Amount: \$19

PolicyCenter creates onset and offset transactions. The onset and offset transactions on the right show that the customer owes \$9 (\$19 minus \$10). If you have enabled the BillingCenter integration, these transactions are sent to BillingCenter. You can also configure the application to send these two transactions to another billing system. The default application calls the billing system when the policy is bound.



The transaction for the (\$10) amount offsets the \$21 amount. Both transactions point to the same cost. The transaction for \$19 points to a new cost.

In the **Policy Transactions** section of the **Summary** screen for the policy change, each line shows the sum of all transactions for each policy transaction.

Summary

Account Information		Period	
Account Number	C000143542	Effective Date	08/13/2013
Account Name	Ray Newton	Expiration Date	02/13/2014
Address	1253 Paloma Ave Floor 0000 Developer Unit Habitation Cube #0000 Arcadia, CA 91007 United States	Term Number	1
County	San Mateo	Total Premium	\$682.00
Address Type	Home	Taxes And Fees	\$49.00
		Total Cost	\$731.00
		Earned Premium	

Underwriter		ig Com. Lines	
Underwriter	Bruce Baker	Acme Low Hazard Insurance	
Primary Named Insured		Associated Policy Transaction	
Name	Ray Newton	Created	08/13/2013
Address	1253 Paloma Ave Floor 0000 Developer Unit Habitation Cube #0000 Arcadia, CA 91007 United States	Closed	08/13/2013
		Type	Policy Change

Completed Policy Transactions							
Compare	Period Eff Date	Trans Eff Date	Trans Close Date	Type	Transaction #	Premium	Comment
<input type="checkbox"/>	08/13/2013	11/13/2013	08/13/2013	Policy Change	745975	\$9.00	
<input type="checkbox"/>	08/13/2013	08/13/2013	08/13/2013	Submission	634621	\$722.00	

Pending Policy Transactions							

Internal Tools for Rating: Financial Transactions Screen

Guidewire provide the **Financial Transactions** screen as an internal tool for use while developing your application. Users with the `internaltools` permission can access this screen. A link to access this screen appears in the **Tools** sidebar of the policy file.

WARNING Guidewire does not support the Internal Tools. Use these tools at your own risk.

The **Financial Transactions** screen for a policy provides links to view:

- All Transactions, Transactions by Job
- Transactions by Period

The following table describes the fields for each transaction.

Field	Description
Eff Date	The effective date.
Exp Date	The expiration date.
Amount	The amount.
Posted Date	The date the cost was posted.
Written	Whether cost is written in the policy. Values are Yes or No.
Charged	Whether the cost has been charged. Values are Yes or No.
ToBeAccrued	Whether there are amounts to be accrued for this cost. Values are Yes or No.
Job Type (Date)	The type of job (policy transaction) that created this cost.
Cost	The type of cost.

The **Transactions by Period** screen lists each period in the policy and allows you to view transactions in two ways: **View by Cost Key** and **View by Cost**.

The **View by Cost Key** choice displays these additional fields:

Field	Description
Cost	The type of cost
Total Charged	Amount that has been charged.
Total Written	Amount written in the policy.

The **View by Cost** choice displays these additional fields:

Field	Description
Cost	The type of cost.
Policy Transaction (job)	The type of policy transaction that created this cost.
Remaining Written Total	The sum of transactions where the Written property is true.
Remaining Charged Total	The sum of transactions where the Charged property is true.



chapter 38

Quote Purging

Over time, the PolicyCenter database accumulates quotes from policy transactions (jobs) not resulting in bound policies and alternate policy periods created through multi-version quoting and side-by-side quoting. As time passes, these policy transactions and policy periods have little business value, increase database storage requirements, and slow response time. Quote purging removes these policy transactions and policy periods from the database.

Note: In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

Quote purging also removes orphaned policy periods, which are policy periods not associated with a policy transaction. Preempted policy transactions result in orphaned policy periods.

Quote purging provides batch processes to remove from the database these policy transactions, policy periods, and other objects. Quote purging is not an end user feature and is only accessible through **Server Tools**.

This topic includes:

- “Quote Purging Overview” on page 441

See also

- “Configuring Quote Purging” on page 435 in the *Configuration Guide*
- “Side-by-side Quoting” on page 159
- “Multi-version Quoting” on page 169
- “Preempted Jobs” on page 503

Quote Purging Overview

Quote purging removes unnecessary policy transactions and policy periods from the PolicyCenter database. You purge quotes by running batch processes.

Quote purging removes policy transactions and other entities from the database. Quote purging is not a reversible operation.

In the default configuration, quote purging provides the following functionality:

- **Purge stale submissions and policy changes** – Purge unbound submissions and policy changes after a specified length of time has passed.
- **Prune unselected versions in a side-by-side quote or multi-version quote** – *Pruning* removes unselected policy periods from submissions and policy changes. Jobs have one selected policy period. In the default configuration, policy transactions can acquire multiple unselected policy periods through side-by-side quoting and multi-version quoting.
- **Purge preempted policy periods** – Preempted policy transactions create *orphaned* policy periods, which are policy periods not associated with a policy transaction. Preempted policy transactions result in orphaned policy periods. Quote purging provides a batch process that removes from the database these orphaned policy periods on submissions and policy changes.
- **Do not purge policies excluded from purging** – Do not purge policy periods on policies flagged as `DoNotPurge`.

Quote purging does not purge policy transactions with archived policy periods. Quote purging does not prune archived policy periods.

IMPORTANT Guidewire does not support configuring quote purging to remove archived policy periods. Do not attempt to implement this functionality without first contacting Guidewire Customer Support for important guidance.

Configuration Business Cases

Quote purging is configurable. Through configuration, some of the types of changes you can make to quote purging are:

- **Change number of days that must pass before purging policy transactions** – Configuration parameters specify the number of days after which a policy transaction is considered for purging. You can change the purge date to affect future runs of the purging batch process.
- **Vary purge date based on business rules** – For example, purge rejected submissions including notes and reasons for the rejection after two years, not the usual six months. If the customer applies for insurance again at a later time, the agent can use the information in the rejected submission in deciding whether to issue a policy. Rejected submissions are identified by rejected underwriting issues.
In addition, the company retains quotes provided by their customer service representatives over the phone for six months, but retains quotes that originate on the website for only four months.
- **Purge other policy transaction types** – In the default configuration, submissions and policy changes are considered for purging. You can add or remove policy transaction types.
- **Purge policy transactions based on business rules** – For example, an insurance company receives inquiries from desirable prospective customers. When an agent enters a submission for a desirable prospect, the agent marks the submission so that the submission will not be purged.
- **Prune unselected versions based on business rules** – For example, an insurance company wishes to retain all versions of side-by-side quotes on policy transactions for existing customers.
- **Remove additional objects such as accounts** – For example, an insurance company takes in a large number of business inquiries. Each new inquiry has an associated account. The company wants to purge the account-related information which clutters the database and slows searches. The insurance company configures quote purging to purge accounts after a year.
- **Collect summary information before purging** – The management team at an insurance company monitors team performance by counting submissions, submissions converted to bound policies, and policy values. The team configures quote purging to retain data from purged policies that is necessary for generating performance metrics.

See also

- “Configuring Quote Purging” on page 435 in the *Configuration Guide*

What Gets Purged or Pruned?

Purging and pruning remove the policy period and all effective dated objects in the policy period branch. Purging and pruning also remove objects directly related to the policy period such as notes, documents, activities, and forms. Purging and pruning does not remove objects not directly related to the policy period.

Because form text data can be shared between policy periods, quote purging does not remove form text data. As a result, form text data can be orphaned as a result of quote purging.

Only purging removes the policy transaction. If purging removes all policy transactions associated with a policy, it also removes the *Policy* object, for example, when a *Policy* only exists because of an unbound submission.

By definition, pruning only removes policy periods associated with alternate versions.

Purging and pruning also remove objects related to underwriting issues on the policy period. In the default configuration, human-touched underwriting issues are not purged or pruned.

See also

- “Objects that Get Purged or Pruned” on page 437 in the *Configuration Guide*
- “List of Batch Processes and Distributable Work Queues” on page 97 in the *System Administration Guide*

Rating Overrides

Rating overrides allow you to manually override the premium that the rating engine automatically generates for a policy. Rating overrides is also referred to as *manual rating* in the insurance industry. After obtaining a quote from the rating engine, you can override rates and amounts, then rate the policy again. Rating overrides allows you to override the base rate and adjusted rate. You can also enter the unprorated amount for the term amount, or a enter a prorated amount to set a flat cost. If a policy has rating overrides, PolicyCenter creates an underwriting issue before releasing the quote. The issue must be approved before certain users, such as agents, can view the quote. This process enables the carrier to verify that overrides are approved before releasing pricing information.

In the base application, the Worker's Compensation, Inland Marine, and Commercial Property lines of business allows rating overrides. You can add rating overrides to other lines.

There are certain types of costs for which rating overrides are not appropriate. You can disable overrides for these costs. For example, you may designate that users cannot override the tax calculation. In the base application, experience modifiers and schedule credits do not allow overrides because the user directly enters values when they are editing the policy. Rating overrides are not appropriate for some inland marine coverages. It is customary to set the manually determined rate on some inland marine risks when entering the risk information, not as an override after rating is run.

This topic includes:

- “Rating Overrides Permissions” on page 445
- “Underwriting Issues for Rating Overrides” on page 446
- “Processing Overrides Across Policy Transactions” on page 446
- “Rating Overrides in the User Interface” on page 447
- “Configuring Rating Overrides” on page 448
- “Adding Rating Overrides to a Line of Business” on page 450

Rating Overrides Permissions

The user must have the correct permissions to view or edit rating overrides. Permissions allow you to view or modify the **Override** screen. Permissions allow you to see messages related to rating overrides on the **Quote** screen.

The following table lists the permissions associated with rating overrides.

Permission	Description
Edit rate and premium overrides	<p>This permission allows you to:</p> <ul style="list-style-type: none"> • Edit or clear overrides on the Rating Overrides screen. • View messages about rating overrides on the Quote screen. • Rate the policy again. <p>In the base configuration, only underwriters have this permission.</p> <p>The code for this permission is <code>editratingoverrides</code>.</p>
View rate and premium overrides	<p>This permission allows you to:</p> <ul style="list-style-type: none"> • View the Rating Overrides screen. • View messages about rating overrides on the Quote screen. <p>In the base configuration, underwriters and other internal users such as policy processors and managers have this permission. Agents and other external users do not have this permission.</p> <p>The code for this permission is <code>vewratingoverrides</code>.</p>

Underwriting Issues for Rating Overrides

If a policy has rating overrides, PolicyCenter creates an underwriting issue of type `QuoteHasManualOverrides` before releasing the quote. The underwriter checks the overrides, then approves the underwriting issue. The user must have an authority grant for this issue type. After the issue is approved, the agent or policyholder can view the quote.

See also

- “Underwriting Authority” on page 411
- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*

Processing Overrides Across Policy Transactions

You can enter rating overrides in any PolicyCenter policy transaction (job). Overrides set for a policy period are preserved in future policy transactions. For example, if you override the rate for liability coverage, that override will be there in a policy change that occurs in the future.

If you create a new policy period, PolicyCenter makes a copy of the policy as of the end of the prior policy period. The end of the prior policy period is the expiration date on renewal and usually the cancellation date for rewrite. The new policy period picks up overrides for that date.

Overrides are not preserved in the following cases:

- **Reinstatement** – Overrides are not preserved if overridden costs occur after the cancellation date. This situation might occur:
 - If the cost of a coverable is overridden and the effective date of the coverable is after the cancellation date.
 - In workers’ compensation policy with an anniversary rating date (ARD). If the cancellation date is before the start of the second policy period, the costs after the cancellation date are deleted at cancellation. When new costs are created on reinstatement, the prior overrides are not preserved.
- **Rewrite** – Whether or not overrides are preserved depends upon the date copied from. If the overridden costs is effective only after or only before this date, then PolicyCenter does not pick up the override.
- **Renewal** – If a renewal policy transaction has rating overrides from the previous policy term that were in effect as of the expiration date of that term. PolicyCenter copies the overrides to the renewal policy transac-

tion, and creates a referral reason for an underwriter. The underwriter decides whether it makes sense to continue the overrides on the new policy term.

- **Anniversary rating date** – Moving the anniversary rating date (ARD) can cause changes to rating overrides on workers' compensation class codes. Be sure to recheck rating overrides after you change the ARD.

Rating Overrides in the User Interface

The following screens display rating override information:

- Rating Overrides on the Quote Screen
- Rating Overrides Screen

Rating Overrides on the Quote Screen

In the **Quote** screen, if there are overrides, PolicyCenter displays this message: **Warning. This quote has some costs overridden.** This screen does not show which values have overrides.

If you have the **View rate and premium overrides** permission, the **Quote** screen contains additional items. The **Policy Premium** tab has an **Override Rating** button which takes you to the **Rating Overrides** screen.

Rating Overrides Screen

The **Rating Overrides** screen allows you to enter overrides to the premiums, taxes, and surcharges. You can view this page if you have the **View rate and premium overrides** permission. To edit this page, you must have the **Edit rate and premium overrides** permission.

The screen has three column sets: **Actual**, **Override**, **Standard**. Each column set has a **Base Rate**, **Adjusted Rate**, **Amount**, and **Term Amount**. (The **Term Amount** column does not apply to the workers' compensation LOB and does not appear.)

The **Standard** column set displays the values from the rating engine and reflects how the row would be rated if there were no overrides. Having the rating engine fill in these values is optional, but it is useful for users to see the impact of an override.

The **Actual** column set displays the values used to calculate the premium. These are the values that display on the **Quote** screen. These are also the values that PolicyCenter sends to the billing application (BillingCenter, for example). The **Basis** field appears only in the **Actual** column set because overrides do not affect it. The values in this column match the values in the **Standard** column set if there are no overrides. If there are overrides, the values match the values in the **Override** column set after rating again.

You can enter an override in the **Override** column. Amounts that can be overridden have text boxes for **Base Rate**, **Adjusted Rate**, **Amount**, or **Term Amount**. You can enter a value in only one of these. You can enter an optional **Reason**.

The screenshot shows the 'Rating Overrides' screen with the following details:

- Header:** Workers' Compensation, Eff. 08/15/2013, Wright Construction, Account # C000212105, Underwriter: Bruce Baker, Under UW Review.
- Buttons:** Rerate, Cancel, Clear All.
- Section:** Standard Premium

Class			Actual				Override				Standard			
Loc.	Code	Description	Base Rate	Adjusted Rate	Basis	Amount	Base Rate	Adjusted Rate	Amount	Reason	Base Rate	Adjusted Rate	Amount	
1	0005	Nurseries—propagation	0.0000	5.0000	500000	\$25,000...			5.0000			5.4540	5.4540	\$27,270.00
	9902	Emp liab increased limits	1.4500	1.4500	25000	\$11,250...						1.4500	1.4500	\$11,250.00

- Section:** Other Premium and Surcharges

Class			Actual				Override				Standard			
Loc.	Code	Description	Base Rate	Adjusted Rate	Basis	Amount	Base Rate	Adjusted Rate	Amount	Reason	Base Rate	Adjusted Rate	Amount	
	9910	Premium discount	-0.0203	-0.0203	36250	(\$736.00)						-0.0203	-0.0203	(\$736.00)
	9911	Expense constant	0.0000	0.0000		\$120.00						0.0000	0.0000	\$120.00
	9912	Terrorism premium	0.0500	0.0500	5000...	\$250.00						0.0500	0.0500	\$250.00
	9950	Tax	0.0700	0.0700	35884	\$2,512.00						0.0700	0.0700	\$2,512.00

In the Nurseries class, the **Base Rate** was previously overridden to 5.25, and the policy was rated again. Therefore, the base rates in the **Actual** and **Override** column sets match. The **Standard** column set displays the original value, 5.4540.

In the Premium discount class, the user has just entered an override to **Amount**. Because the user has not yet clicked the **Rerate** button, the values in the **Actual** and **Standard** column sets are the same.

The following table describes some of the fields on the **Rating Overrides** screen.

Field	Description
Base Rate	The rate from your rating table.
Adjusted Rate	The rate after applying discounts or other adjustments to the base rate.
Term Amount	The unprorated amount for the policy term.
Amount	The amount calculated from the adjusted rate and basis. Enter a value in this field to specify a flat amount.
Note: This value is not prorated automatically even if the policy is later changed.	

The following table describes some of the buttons on the **Rating Overrides** screen.

Field	Description
Rerate	This button sends the updated data to the rating engine.
Clear All	Use this button to clear all overrides. Remove overrides individually by clearing the text entry fields in the Override column.

Configuring Rating Overrides

This topic provides step-by-step instructions on how to override premiums.

To override a rating

- On the Quote page, click the **Override Rating** button.

The **Rating Overrides** screen appears.

2. Enter a value in the **Adjusted Rate** field. In this example, enter 4.0 in the **Adjusted Rate** for **NURSERIES**.

Optionally, enter a reason in the **Reason** field.

Rating Overrides (Return to Quote)													
			Actual				Override				Standard		
Loc.	Code	Description	Base Rate	Adjusted Rate	Basis	Amount	Base Rate	Adjusted Rate	Amount	Reason	Base Rate	Adjusted Rate	Amount
1	0005	NURSERIES - propagation and cultivation of nursery stock	5.4540	5.4540	350000	\$19,089.00		4.0	\$	Low risk	5.4540	5.4540	\$19,089.00
	9902	Emp liab increased	1.4500	1.4500	19089	\$8,590.00			\$		1.4500	1.4500	\$8,590.00

3. Click the **Rerate** button.

This action sends the new values off to the rating engine.

The **Quote** screen appears and displays a message that there are overrides. This warning appears only if you have permission to view overrides.

The **Quote** screen displays the values that the rating engine returned. The override you made to the nurseries class code affects other values on this screen. The override affects the value of the **Manual Premium**. The **Manual Premium** is used to calculate the remaining values in the **Standard Premium** table. The **Standard Premium** value is input into the **Other Premium and Surcharges** table.

Quote																																																									
< Back Next > Release Lock Edit Submission New Version Save Draft Bind Options Close Options																																																									
Warning. This quote has some costs overridden.																																																									
<table> <tr> <td>Submission Number</td> <td>37339</td> <td>Total Premium</td> <td>\$20,335.00</td> </tr> <tr> <td>Policy Period</td> <td>10/20/2009 - 10/20/2010</td> <td>Taxes & Surcharges</td> <td>\$1,423.00</td> </tr> <tr> <td>Primary Named Insured</td> <td>Wright Construction</td> <td>Total Cost</td> <td>\$21,758.00</td> </tr> <tr> <td>Policy Address</td> <td>846 Yount Ln. Floor 0000 Developer Unit Habitation Cube #0000 Hollywood, CA 91357</td> <td></td> <td></td> </tr> <tr> <td>County</td> <td>San Mateo</td> <td></td> <td></td> </tr> <tr> <td>Country</td> <td>United States of America</td> <td></td> <td></td> </tr> <tr> <td>Address Type</td> <td>Home</td> <td></td> <td></td> </tr> <tr> <td>Address Description</td> <td>Created by the Address Builder with code 0</td> <td></td> <td></td> </tr> </table>											Submission Number	37339	Total Premium	\$20,335.00	Policy Period	10/20/2009 - 10/20/2010	Taxes & Surcharges	\$1,423.00	Primary Named Insured	Wright Construction	Total Cost	\$21,758.00	Policy Address	846 Yount Ln. Floor 0000 Developer Unit Habitation Cube #0000 Hollywood, CA 91357			County	San Mateo			Country	United States of America			Address Type	Home			Address Description	Created by the Address Builder with code 0																	
Submission Number	37339	Total Premium	\$20,335.00																																																						
Policy Period	10/20/2009 - 10/20/2010	Taxes & Surcharges	\$1,423.00																																																						
Primary Named Insured	Wright Construction	Total Cost	\$21,758.00																																																						
Policy Address	846 Yount Ln. Floor 0000 Developer Unit Habitation Cube #0000 Hollywood, CA 91357																																																								
County	San Mateo																																																								
Country	United States of America																																																								
Address Type	Home																																																								
Address Description	Created by the Address Builder with code 0																																																								
This quote will require underwriting approval prior to binding.																																																									
<table> <tr> <td>Policy Premium</td> <td>Override Rating</td> </tr> <tr> <td>Standard Premium</td> <td></td> </tr> </table>												Policy Premium	Override Rating	Standard Premium																																											
Policy Premium	Override Rating																																																								
Standard Premium																																																									
<table> <tr> <td>Loc.</td> <td>Code</td> <td>Description</td> <td>Basis</td> <td>Rate</td> <td>Amount</td> </tr> <tr> <td>1</td> <td>0005</td> <td>NURSERIES - propagation and cultivation of nursery stock</td> <td>350000</td> <td>4</td> <td>\$14,000.00</td> </tr> <tr> <td></td> <td></td> <td>Manual Premium</td> <td></td> <td></td> <td>\$14,000.00</td> </tr> <tr> <td></td> <td>9902</td> <td>Emp liab increased limits</td> <td></td> <td>14000</td> <td>1.4500</td> <td>\$6,300.00</td> </tr> <tr> <td></td> <td></td> <td>Subject Premium</td> <td></td> <td></td> <td></td> <td>\$20,300.00</td> </tr> <tr> <td></td> <td></td> <td>Modified Premium</td> <td></td> <td></td> <td></td> <td>\$20,300.00</td> </tr> <tr> <td></td> <td></td> <td>Standard Premium</td> <td></td> <td></td> <td></td> <td>\$20,300.00</td> </tr> </table>												Loc.	Code	Description	Basis	Rate	Amount	1	0005	NURSERIES - propagation and cultivation of nursery stock	350000	4	\$14,000.00			Manual Premium			\$14,000.00		9902	Emp liab increased limits		14000	1.4500	\$6,300.00			Subject Premium				\$20,300.00			Modified Premium				\$20,300.00			Standard Premium				\$20,300.00
Loc.	Code	Description	Basis	Rate	Amount																																																				
1	0005	NURSERIES - propagation and cultivation of nursery stock	350000	4	\$14,000.00																																																				
		Manual Premium			\$14,000.00																																																				
	9902	Emp liab increased limits		14000	1.4500	\$6,300.00																																																			
		Subject Premium				\$20,300.00																																																			
		Modified Premium				\$20,300.00																																																			
		Standard Premium				\$20,300.00																																																			
Other Premium and Surcharges																																																									
<table> <tr> <td>Loc.</td> <td>Code</td> <td>Description</td> <td>Basis</td> <td>Rate</td> <td>Amount</td> </tr> <tr> <td>9910</td> <td>Premium discount</td> <td>20300</td> <td>-0.0128</td> <td>(\$260.00)</td> <td></td> </tr> <tr> <td>9911</td> <td>Expense constant</td> <td></td> <td></td> <td></td> <td>\$120.00</td> </tr> <tr> <td>9912</td> <td>Terrorism premium</td> <td>350000</td> <td>0.0500</td> <td></td> <td>\$175.00</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>\$20,335.00</td> </tr> <tr> <td>9950</td> <td>Tax</td> <td>20335</td> <td>0.0700</td> <td></td> <td>\$1,423.00</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>\$21,758.00</td> </tr> </table>												Loc.	Code	Description	Basis	Rate	Amount	9910	Premium discount	20300	-0.0128	(\$260.00)		9911	Expense constant				\$120.00	9912	Terrorism premium	350000	0.0500		\$175.00						\$20,335.00	9950	Tax	20335	0.0700		\$1,423.00						\$21,758.00				
Loc.	Code	Description	Basis	Rate	Amount																																																				
9910	Premium discount	20300	-0.0128	(\$260.00)																																																					
9911	Expense constant				\$120.00																																																				
9912	Terrorism premium	350000	0.0500		\$175.00																																																				
					\$20,335.00																																																				
9950	Tax	20335	0.0700		\$1,423.00																																																				
					\$21,758.00																																																				
Total Cost: \$21,758.00																																																									

4. Click the **Override Rating** button. The **Rating Overrides** screen appears. Notice that the **Adjusted Rate** in the **Actual** and **Standard** columns display different values. The **Standard** column displays the rate without override. The **Actual** column displays the rate with the override.

Rating Overrides (Return to Quote)													
			Rerate			Cancel			Clear All				
Standard Premium													
Class			Actual			Override			Standard				
Loc.	Code	Description	Base Rate	Adjusted Rate	Basis	Amount	Base Rate	Adjusted Rate	Amount	Reason	Base Rate	Adjusted Rate	Amount
1 0005	NURSERIES	- propagation and cultivation of nursery stock	0.0000	4.0000	350000	\$14,000.00			\$ 13000	Low risk	5.4540	5.4540	\$19,089.00
9902	Emp liab increased		1.4500	1.4500	14000	\$6,300.00			\$		1.4500	1.4500	\$6,300.00

5. Remove the value in the **Adjusted Rate** column.
 6. Enter a value in the **Amount** column and rerate the policy. In this example, enter 13000 for the nurseries class code.
 7. Click the **Rerate** button.

The **Quote** page appears again. Because you entered a flat amount, the **Basis** and **Rate** fields are empty for the nurseries class code.

Adding Rating Overrides to a Line of Business

In the default configuration, rating overrides are provided in the Workers' Compensation, Inland Marine, and Commercial Property lines of business. Follow these steps to add rating overrides to a new or existing line of business:

1. Enabling the Override Rating Button
2. Creating a Panel Set for Rating Overrides
3. Updating the Rating Engine to Handle Overrides

Enabling the Override Rating Button

Enable the **Override Rating** button on the **Quote** screen. The visibility of this button is controlled in part by a policy line boolean property called **SupportsRatingOverrides**. The default implementation in **PolicyLineMethodsDefaultImpl.gs** returns **false**. On the **LOBPolicyLineMethods** class, override the property to return **true**.

1. In Studio, navigate to **configuration** → **gsrc**. Open **gw.lob.LOB.LOBPolicyLineMethods.gs**.
 2. Add the following code to override the **SupportsRatingOverrides** property. Set the value to **true**. (Or modify the code if it already exists.)
- ```
override property get SupportsRatingOverrides() : boolean {
 return true
}
```

### Creating a Panel Set for Rating Overrides

To do rating overrides, you must create a screen for viewing and editing overrides in the line of business. Follow these steps to create a panel set for a **Rating Overrides** screen.

1. In Studio, navigate to **configuration** → **config** → **Page Configuration** → **pcf** → **line** → **LOB**.
2. Right-click the **LOB** and select **New** → **PCF Folder**.

3. In the dialog box, enter **ratingoverride** in **Folder Name**. Click **OK**.
  4. Right-click the **ratingoverride** folder and select **New → PCF File**.
  5. In the dialog box, enter **RatingOverride**. In the **File type** list, select **Panel Set**. In the **Mode** text box, enter **LOBLine**. Click **OK**.
- Studio creates a PCF file named **RatingOverridePanelSet.LOBLine** in the **ratingoverride** folder.
6. Design the layout of the panel set.

You can base this panel set on the **Rating Overrides** screens from one of the policy lines that provides rating overrides. In these lines of business, only the cost fields with names that begin with **Override** are editable, and then only if the individual cost row is flagged as **Overridable**. The **Actual** and **Standard** fields are always read-only.

## Updating the Rating Engine to Handle Overrides

Modify your rating engine to handle overrides in the policy line. This topic provides general guidance on how to modify the rating engine provided in the default configuration.

The user adds overrides in the **Rating Overrides** screen. PolicyCenter passes the costs, which contain the current values of the override fields, to the rating engine. It is up to the rating engine to check whether there is an override, and then use it in the calculation of the new cost. For example, if a term amount is overridden, then the rating engine skips the calculation of the basis and rate, and simply sets this term amount on the cost. The rating engine uses that amount for later rating steps (such as calculating a subtotal for taxes).



# Document Management

PolicyCenter creates and manages documents associated with policies and accounts. These documents can either be online—existing in or created in PolicyCenter—or printed documents. For example, you can write and send the insured a letter to acknowledge a new submission. The insured might email you documents related to the safety of the insured location or garage.

Use the Document feature in PolicyCenter to:

- Create new documents from templates.
- Have another user approve a document you wrote before it is sent.
- Store documents, both those you create and those received from other sources.
- Search for documents.
- Link to external documents.
- Remove documents.
- Associate a document with a single policy or account.
- Associate the creation of a document with an activity.
- Create and send a document to perform a task for an activity.
- Create and send a document from rules or workflows.
- Extend these default capabilities by integrating to an external document management system (DMS).

By default, PolicyCenter stores documents as files in your local file system. You manage these files with directory and file commands. You can also integrate documents with an external document management system. For more information, see “Configuring Document Management” on page 457.

**Note:** All documents must either be ASCII or use the UTF-8 character set.

This topic includes:

- “Document Security” on page 454
- “Working with Documents” on page 454
- “Configuring Document Management” on page 457
- “Document Management Integration” on page 459

**See also**

- “Document Creation” on page 105 in the *Rules Guide*
- “Document Management” on page 265 in the *Integration Guide*

## Document Security

PolicyCenter provides a set of system permissions to provide security for all documents as seen in the following table. You can also use these permissions to define different security types for documents and assign permissions to users that relate to these security types.

**Note:** See “Access Control for Documents and Notes” on page 652 for more information.

### Permissions Related to Documents

The following system permissions provide security for documents.

| Name         | Purpose of permission                                          |
|--------------|----------------------------------------------------------------|
| doccreate    | Add documents to a policy, account, or policy transaction      |
| docdelete    | Remove documents from a policy, account, or policy transaction |
| docedit      | Edit documents on a policy, account, or policy transaction     |
| docmodifyall | Modify any document, regardless of security type               |
| docview      | View documents on a policy, account, or policy transaction     |
| docviewall   | View any document, regardless of its security type             |

## Hidden Documents

Hiding documents is a way to remove obsolete documents from your list of documents without deleting them. When you hide a document, you no longer see it listed in the **Documents** screen unless you indicate that want to see hidden documents.

You can make a document hidden by setting **Hidden** to Yes on the **New Document** or **Document Details** screen. Setting **Hidden** to Yes sets the **Obsolete** flag on Document and does not retire the document in the database. You can view hidden documents by setting **Include Hidden Documents** to Yes on the **Documents** screen.

Hiding a document is not the same as deleting it. The **docdelete** permission is necessary to fully delete documents. Only users who have that permission can delete. Since document deletion is at least semi-permanent, you can delete only one document at a time through the user interface.

## Working with Documents

This topic describes the various ways you can work with documents and includes:

- “Viewing Account and Policy Documents” on page 455
- “Searching for Documents” on page 455
- “Adding a New Document” on page 455
- “Linking to an Existing Document” on page 455
- “Creating a New Document” on page 456
- “Editing a Document” on page 456
- “Removing a Document” on page 456
- “Using an Activity to Create a Document” on page 456

- “Creating a Document with a Rule or in a Workflow” on page 457
- “Relating a Document to a PolicyCenter Entity” on page 457
- “Creating a Document Template” on page 457

Select the **Documents** link in the left sidebar to perform any of these actions on existing documents from any policy or account screen. To work with a new document, select **New Document** from the **Actions** menu while in any policy or account screen.

## Viewing Account and Policy Documents

The **Documents** screen shows documents for accounts and policies. To open this screen, click the **Documents** link in the left sidebar of all account or policy screens.

The bottom of this screen initially displays the unfiltered list of all documents. For policies, this screen lists all documents associated with the policy. For accounts, this screen lists all documents associated with the account and all policies on the account. Use the search pane at the top of the screen to filter the list of documents.

You can view all documents for which you have permission. Click any document name to view its details.

The `RestrictSearchesToPermittedItems` search parameter in the `config.xml` file determines whether you can see a document in the list that you do not have permission to view.

## Searching for Documents

Use the **Search** pane of the **Documents** screen to search for documents. This pane includes the search attributes that follow. You can use the following search parameter values for a document after you create the document or link to it:

- **Document Name** – Especially useful for locating hard-copy documents.
- **Status** – From the `DocumentStatusType` typelist. You can set this value in the user interface, but its main use is to track the approval process of a document in rules.
- **Date Range** – Set the start and end of a date range.
- **Author** – The name of the creator, the sender, or some other value.
- **Include Hidden Documents** – Whether to include Hidden Documents in the search.

### See also

- “Configuring PolicyCenter Search” on page 332 in the *Configuration Guide*

## Adding a New Document

Select **New Document** from the **Actions** menu while in any policy or account screen to see the following choices for adding documents to the current policy:

- **Link to an existing document** – See “Linking to an Existing Document” on page 455.
- **Create a new document from a template** – See “Creating a New Document” on page 456.

## Linking to an Existing Document

1. Select **Actions** → **New Document** → **Link to an existing document**.
2. Browse to the location of your document.
3. Click **Update** to create the link.

## Creating a New Document

1. Select **Actions** → **New Document** → **Create a new document from a template**.
2. In the **New Document** panel, select the template to use. Initially, leave the field empty and click the **Select Template** search icon. You cannot create a document without specifying an existing template.

After attaching a document for a policy, but not an account, PolicyCenter automatically shows only the appropriate document types for that type of policy. This behavior is based on the configuration settings in the Product Model for that policy type.

After clicking the search icon, a second screen opens, enabling you to search for document templates as follows:

  - a. The pane displays a list of document templates at the bottom. Find the one you want and click **Select**.
  - b. If there are too many to choose from, limit the list by using the search pane.

For example, choose a document type from the **Type** picker and click **Search**. The list of document templates displays again with a list showing only that document type. After you find your choice, click **Select**.

After you select a template, PolicyCenter displays numbered steps along the left side of the screen.

**Note:** The base configuration Sample Acrobat document, `SampleAcrobat.pdf`, uses Helvetica font. If you intend to create a document that uses Unicode characters, such as one that uses an East Asian language, the document template must support a Unicode font. Otherwise, the document does not display Unicode characters correctly.

3. Follow the steps on the screen. You can also add additional criteria from the right side of the screen.

The file attributes, used by your document management system, need not be the same as the comparable object values that appear in the document.

The document appears in its native editor. If you edit the document, remember to save it.

4. Click **Update** to save your work.

After you create the document, you can take additional steps, such as sending this document as an email attachment. You can also print it and send it through the mail. Additionally, you can use any features provided by your document management system.

## Editing a Document

You can edit a file in the **Documents** screen.

You can edit documents if you have sufficient permissions or if the document does not have a status of **Final**. To edit a document, click **Edit** in the **Actions** column for the document. PolicyCenter enables the **Upload** and **Discard** buttons. Clicking **Upload** returns the document to your document management system. Clicking **Discard** discards your local edits.

## Removing a Document

After locating the document in the **Documents** list, click **Delete**. If no button is visible in the **Delete** column, you do not have the authority to delete that file. See “Document Security” on page 454.

## Using an Activity to Create a Document

If a document template is specified in the activity pattern of an activity, all activities created from that pattern have a **Create Document** button visible after the activity opens. Clicking this button displays a popup window identical to the **Create New Document** version of a **New Document** screen. You can create the document. Since an activity pattern can indicate only one template, any activity creates only one type of document.

For the **New Document** screen, see the example in “Adding a New Document” on page 455.

**Note:** In the default configuration, no activity pattern specifies a document template.

## Creating a Document with a Rule or in a Workflow

To learn how to automatically create documents by using rules, see “Generating Documents from Gosu” on page 296 in the *Integration Guide*. Use similar rules to create a document in a workflow.

## Relating a Document to a PolicyCenter Entity

You can relate a document to a policy, account, or policy transaction.

When creating a document, you can specify the entity in the **Related To** field. If the document already exists, you create this relationship by using the **Related To** field after linking the document to a policy, account, or policy transaction. The **Related To** relationship can only have one value for any given document.

## Creating a Document Template

A document template consists of two files. One file is a document template descriptor file, which contains the metadata, such as its name, ID, and MIME type. The other file is the document template itself, which contains the document contents.

You can view and edit the document templates and descriptors by navigating to **configuration** → **config** → **resources** → **doctemplates** in Studio.

Document template files are in the following directory:

`PolicyCenter/modules/configuration/config/resources/doctemplates`

- There are several example files in that directory. The best way to create a new template is to generate this pair of files from copies of these examples. The descriptor file is in XML format. Studio does not contain a special editor to help generate new templates.
- For details about document management, document templates, and related integration points, see “Document Management” on page 265 in the *Integration Guide*.

## Configuring Document Management

The following configuration parameters in the `config.xml` file control the display and editing of files in a document management system. The suggested values assume that you are using the file system on the application server.

**IMPORTANT** Guidewire strongly suggests integrating with an external document management system rather than using this default system. The default system does not support versioning, sending documents, providing security, and so on.

| Parameter                                                                                  | Description                                                                                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>AllowDocumentAssistant</code><br><code>UseDocumentAssistantToDisplayDocuments</code> | Set both to <code>true</code> so that the Document Assistant ActiveX control displays document contents. These settings are necessary to display and edit documents. See “Guidewire Document Assistant Configuration Parameters” on page 117 in the <i>System Administration Guide</i> .         |
| <code>DisplayDocumentEditUploadButtons</code>                                              | Set to <code>true</code> to display <b>Edit</b> and <b>Upload</b> buttons, which enable editing and returning edited files to the document management system. <code>AllowDocumentAssistant</code> must also be <code>true</code> , and <code>IDocumentContentSource</code> must support editing. |

| Parameter                             | Description                                                                                                                                                                                                                                                                                             |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DocumentContentDispositionMode        | How to display a retrieved document. Set to <code>inline</code> to have it appear in a browser. Set to <code>attachment</code> to have it opened by its editor, as determined by its MIME type. If <code>inline</code> , the <code>AllowDocumentAssistant</code> parameter must be <code>false</code> . |
| DocumentTemplateDescriptorXSDLocation | The location of the XSD file that validates document template descriptor files, usually <code>document-template.xsd</code> . See “ <code>DocumentTemplateDescriptorXSDLocation</code> ” on page 48 in the <i>Configuration Guide</i> for a description of how to set this location.                     |
| MaximumFileUploadSize                 | Because documents either reside on the application server or are uploaded to the document management system by the server, this value can protect the server. The default is 20 MB.                                                                                                                     |
| MaxDocTemplateSearchResults           | The maximum number of template search results to show before showing a warning in the user interface. The default is 50. The warning appears if the limit is exceeded.                                                                                                                                  |

Another section of the `config.xml` file maps document MIME types to file extensions and icons in the user interface:

```
<mimetypemapping>
 <mimetype name="application/msword"
 extensions=".doc"
 icon="word.gif"
 description="Microsoft Word Document"/>
 <!-- more mappings -->
</mimetypemapping>
```

#### See also

- To configure search parameters for documents, see “Searching for Documents” on page 455.
- For details about document management and related integration points, see “Document Management” on page 265 in the *Integration Guide*.

## Document Management Integration

The following are the main plugin interfaces used to integrate with a document management system. Each plugin interface has a default plugin implementation class.

Interface	Description
IDocumentMetadataSource	<p>PolicyCenter passes search parameters—metadata—to this interface, which searches its metadata and returns a list of documents found. You can use it to interface with a system for storing document metadata—name, id, status, author, and so on. If the system specifies none, then the PolicyCenter database stores the metadata. This interface is separate from IDocumentContentSource because of different architectural requirements.</p> <p>In the base configuration, this interface is implemented by the class <code>gw.plugin.document.impl.LocalDocumentMetadataSource</code>.</p>
IDocumentContentSource	<p>PolicyCenter passes this interface metadata for one file, which returns its content. This interface:</p> <ul style="list-style-type: none"><li>• Interfaces with a document storage system.</li><li>• Contains methods for creating, updating, and retrieving document contents.</li><li>• Supports the following document retrieval modes:<ul style="list-style-type: none"><li>• Document contents.</li><li>• Web page containing an ActiveX control (see “Working with Documents” on page 454).</li><li>• Gosu executed by client rules.</li><li>• URL to a local content store.</li></ul></li></ul> <p>In the base configuration, this interface is implemented by the class <code>gw.plugin.document.impl.AsyncDocumentContentSource</code>.</p>
IDocumentProduction	<ul style="list-style-type: none"><li>• Interface to a document creation system.</li><li>• Document creation process can involve extended workflow and/or asynchronous processes.</li><li>• Can depend on or set document fields.</li></ul> <p>In the base configuration, this interface is implemented by the class <code>gw.plugin.document.impl.LocalDocumentProductionDispatcher</code>.</p>
IDocumentTemplateSource and IDocumentTemplateDescriptor	<ul style="list-style-type: none"><li>• Basic interfaces for searching for and retrieving templates describing the document to be created.</li><li>• Include basic metadata (name, MIME type, and so on) and a pointer to the template content.</li><li>• Specifically, the <code>IDocumentTemplateDescriptor</code> plugin describes the templates used to create documents and the <code>IDocumentTemplateSource</code> plugin actually lists and retrieves the document templates.</li></ul>

For more information about document management and related integration points, see “Document Management” on page 265 in the *Integration Guide*.



# Policy Forms

For the insured customer, the physical representation of an insurance policy is a collection of *policy forms*. Policy forms define aspects of the policy such as coverages, exposures, exclusions, and government regulations.

This topic includes:

- “Forms Overview” on page 461

## Forms Overview

For the insured customer, the physical representation of an insurance policy is a collection of policy forms. Policy forms define aspects of the policy such as coverages, exposures, exclusions, and government regulations. PolicyCenter supports viewing a list of forms in the user interface. You can also integrate with a forms printing system hosted separately from PolicyCenter. Although printing forms is primarily associated with issuance in a submission policy transaction, forms can be printed as part of any policy transaction. For example, a policy change might trigger reprinting a changed form, or printing additional forms that are now necessary because of newly-added vehicles or other changes.

**Note:** For what Guidewire calls policy forms, the insurance industry sometimes calls *endorsements* on a policy. Guidewire avoids the term endorsements due to its ambiguity in the industry, since endorsements sometimes refer to policy changes.

All PolicyCenter forms are automatically inferred forms, not manually added forms. In PolicyCenter, forms are not added explicitly by the user. Instead, PolicyCenter users add coverages, exclusions, and other policy data. Then PolicyCenter generates forms automatically by using forms inference logic and configuration settings. In addition, the form itself never contains variable information that is not already encoded in the data model or product model for that policy.

If a user submits a new auto policy, the forms to print when issuing the policy can be inferred by the coverages, vehicles, and other fields. If the insured later adds another vehicle to a policy, PolicyCenter determines which forms to reprint and whether to print entirely new forms for the new vehicle.

In the user interface, forms are listed on the **Forms** screen after the policy is quoted. The user interface displays a list of forms not the actual representation of the forms. Before binding, the list of forms is just a preview, not the

final list of forms that will be attached to the policy. After the policy is bound, the list of forms may be different. The list may be different because the information to accurately infer some forms is available only at binding time. When the policy is issued, your integration code sends XML data describing the form to an external system which prints the forms on paper or in an electronic format.

The forms feature of PolicyCenter has the following components:

What	Where to configure	Description
Forms basic definition	In PolicyCenter, go to <b>Administration → Policy Forms</b> .	Defines forms that can be inferred for policies created in PolicyCenter.
Custom inference classes	Custom Gosu classes defined in Studio.	You can define custom inference classes to get more advanced behavior than is possible with the basic forms definitions. These custom classes define the conditions that determine when to add the form to the policy.
Forms preview	No configuration needed.	The job (policy transaction) wizard user interface displays a preview of the list of forms for the current policy.
Form printing integration	Event Fired rules. Custom messaging plugins.	Event Fired rules intercept forms issuance events and generate messages. Custom messaging plugins (destinations) that you register must send the XML payload to the forms printing system. Typically, this occurs only prior to binding a policy transaction.

If your forms printing system integrates with a document management system (DMS), the printing system can generate a visual representation of the form and add it to the DMS. After it does this, the integration code can also connect with PolicyCenter to let it know there is a new document associated with the policy. You can then view the policy from the **Documents** screen in the policy file.

#### See also

- “Policy Form Pattern Administration” on page 683 for more information about how to administer forms.
- “Forms Integration” on page 477 in the *Integration Guide* for more information about inference classes and forms printing integration.
- “Document Management” on page 265 in the *Integration Guide* for more information about document management.



## chapter 42

# Policy Data Spreadsheet Import/Export

In PolicyCenter, policy data spreadsheet import/export enables you to export policy data to and from a spreadsheet. You can review and revise the exported data in a spreadsheet editor. You can import data from a spreadsheet into PolicyCenter.

You can use policy data spreadsheet import/export to review or enter large amounts of data for commercial policies. You can review existing policy data in a spreadsheet, add or update the data, then import that data into PolicyCenter.

Policy data spreadsheet import/export uses the Office Open XML Workbook (.xlsx) spreadsheet format.

With policy data spreadsheet import/export you can:

- Export a template to a spreadsheet. The template provides just the column headings and typelists for fields needed in new submission policy transactions.
- Export policy data to a spreadsheet. The spreadsheet provides a snapshot of a current policy transaction. You can use this snapshot for review purposes or to make modifications for most policy transactions, including submission, change, renewal, and rewrite.
- Import updated or newly added policy data from a spreadsheet into PolicyCenter. Prior to committing the import, you can preview the changes that the import operation will make to the policy, and then accept or reject the entire import operation.
- Configure export formats that specify the fields to export within each supported coverable.
- Extend this functionality to handle spreadsheet import/export for additional coverables and other lines of business.

In the base configuration, policy data import/export is implemented for buildings and locations in the commercial property line of business.

This topic includes:

- “Large Policy Workflow Using Policy Data Spreadsheet Import/Export” on page 464
- “Policy Data Spreadsheet Import/Export in Commercial Property” on page 464
- “Using Spreadsheets Generated by Policy Data Spreadsheet Import/Export” on page 465

### See also

- “Administering Policy Data Spreadsheet Import/Export” on page 697
- “Configuring Policy Data Spreadsheet Import/Export” on page 483 in the *Configuration Guide*

## Large Policy Workflow Using Policy Data Spreadsheet Import/Export

Using policy data spreadsheet import/export, you can directly import policy details directly from a standard spreadsheet. To ensure the format and structure of the captured data, you can export both a template and a full spreadsheet from PolicyCenter. The template contains column headings only and is used as a basis for capturing data for new coverages. The full spreadsheet contains both column headings and existing policy data for a specific policy transaction. You can use the full spreadsheet for reviewing a policy and for capturing additions, deletions, and changes to the details of existing coverages. You can use this spreadsheet to manipulate a large number of changes in a bulk operation. For example, you can add 10% to a certain coverage term in the spreadsheet, and then import the change back into the policy transaction. Furthermore, PolicyCenter imposes no arbitrary restrictions on the number of coverables and coverages it can track. Therefore, there is no need to keep a paper trail as a separate system of record for any part of a policy.

In the base configuration, policy data spreadsheet import/export exports a representative set of fields for the locations and buildings in the Commercial Property line of business. You can extend Policy data spreadsheet import/export to include additional fields as needed, including fields that have been added in specific PolicyCenter implementations. You can also extend Policy data spreadsheet import/export to provide similar capabilities in other lines of business.

Within the set of fields that have been enabled for export, an administrator with appropriate permissions can define formats that remove selected fields (columns) from individual export operations. You can define the format so that it exports spreadsheets that contain exactly the data you want to review or capture in a particular policy transaction.

Each exported spreadsheet is identified with a single PolicyCenter policy transaction. As a general rule, the spreadsheet is only imported into the same policy transaction or a policy transaction (job) whose `basedOn` property leads back to the exported policy transaction. PolicyCenter displays an error if you attempt to import a spreadsheet into an incompatible policy transaction. To help you match a spreadsheet to a policy transaction, the file name that PolicyCenter suggests contains the policy number, policy transaction type, transaction number, and date. Although you may change the file name when saving an exported spreadsheet, Guidewire recommends that you retain the transaction number for ease of matching spreadsheets to policy transactions.

After completing an import operation, PolicyCenter can save a log file. The log file contains information such as the number of coverables read, added, changed, or removed. You can consult this log file if there are errors on import. The log file identifies the spreadsheet row and column where each error occurred.

## Policy Data Spreadsheet Import/Export in Commercial Property

With policy data spreadsheet import/export, you can export to and import coverables from PolicyCenter to the Office Open XML Workbook (.xlsx) spreadsheet format. Assuming you have the necessary permissions, you can perform the following operations within a submission, change, renewal, or rewrite policy transaction:

- **Export a locations template or a buildings template to a spreadsheet.** The exported template spreadsheet contains column headings only and is used to capture data for new submissions. Using a spreadsheet program, you can enter new data, one row per location or building.
- **Export locations or buildings to a spreadsheet.** Locations and Buildings spreadsheets are pre-filled with data describing all existing coverables of a given policy. You can use exported spreadsheets to review policy data or can be edit to add, remove, or change coverables within a policy transaction.
- **Import locations or buildings from a spreadsheet.** Coverables can be imported into the same policy transaction from which they are exported, or into a related policy transaction. A related policy transaction is one whose `basedOn` property leads back to the policy transaction that was actually exported. PolicyCenter attempts to find the matching entities in the graph and change those entities using the data in the spreadsheet you are importing. Before making any changes, PolicyCenter displays a warning message. By matching

policy transactions using the property, you can export a submission policy transaction and import the result into a policy change based on the exported submission.

For ease of matching a spreadsheet with a policy transaction, the transaction number is part of the file name that PolicyCenter proposes when saving the exported spreadsheet. During import operations, you can preview changes before accepting them. After importing, you can correct any validation errors just as if the data had been entered interactively into PolicyCenter.

- **Use custom export formats.** When exporting either a template or a coverables spreadsheet, you can select an export format that exports only the specific data columns to capture the needed details. Administrators can define new export formats.

You can extend commercial property policy data spreadsheet import/export in the following ways:

- Customize column headings, including translating headings into other languages.
- Include additional coverage details within the commercial property line of business

You can extend policy data spreadsheet import/export using Guidewire Studio, Gosu code, and an XML editor. You define the fields to export from each coverable in an XML file, along with the column headings that appear in the spreadsheet. Spreadsheet column headings are defined as separate attributes. Therefore, they need not match the field names they represent and can be translated as needed for various locales.

## Using Spreadsheets Generated by Policy Data Spreadsheet Import/Export

This topic provides step-by-step instructions for working with spreadsheets exported by the policy data spreadsheet import/export. The examples demonstrate this feature in the commercial property line of business.

**Note:** Policy data spreadsheet import/export uses the Office Open XML Workbook (.xlsx) spreadsheet format. You can use spreadsheet import/export with a legacy version of Microsoft Excel (Microsoft Office 2003 or prior version). Install a free compatibility pack from Microsoft that enables you to read and write .xlsx files.

### Exporting a Spreadsheet from PolicyCenter

This topic provides step-by-step instructions for exporting buildings and locations from a policy in PolicyCenter to a spreadsheet.

#### To export a spreadsheet

1. Start a policy transaction in the commercial property line. You can export a spreadsheet during a submission, change, or renewal. Advance to the **Buildings and Locations** screen.
2. In the **Buildings and Locations** wizard step, click **Spreadsheet** and select **Export** from the drop-down list. The **Export to Spreadsheet** screen appears.

**3.** Make selections as described in the following table.

<b>Export</b>	<ul style="list-style-type: none"> <li>• <b>Commercial Property Locations</b> – Exports a spreadsheet that enables you to add new locations.</li> <li>• <b>Commercial Property Buildings</b> – Exports a spreadsheet that enables you to add buildings. If needed, you can also specify new locations for the new buildings. The Commercial Property Buildings spreadsheet is useful because you can add both buildings and their locations in a single operation. PolicyCenter validation requires that each location have at least one building.</li> </ul>
<b>All from this version</b>	<p>Exports a spreadsheet that contains all existing coverables, enabling you to make changes to existing policy data and add new coverables.</p> <p>You can only import this spreadsheet into the same policy transaction from which it was exported.</p>
<b>Template for any policy transaction</b>	<p>Exports a template spreadsheet that contains only column headings, enabling you to add new coverables only.</p> <p>You can use this spreadsheet to import new policy data into any policy transaction.</p>
<b>Format</b>	<p>Lists the export formats that have been defined by an administrator. Each export format defines a subset of fields to export. To export all available fields, select <b>All Available</b>. To export a subset of fields, select the corresponding format from this list.</p> <p>Consult the person who designed the export formats to determine the appropriate formats to use for various situations.</p>
<b>Language</b>	<p>Lists available languages based on the installed PolicyCenter language packs. Select the language that appears in the exported spreadsheet column headings.</p>

**4.** Click **Export to Spreadsheet** to export the specified spreadsheet. Specify a location and file name for the spreadsheet. PolicyCenter provides recommendations for filenames to help identify the transaction number and date for future reference.

**5.** Use a compatible spreadsheet program to open the exported spreadsheet and fill in the fields.

If you exported a **Template for any policy transaction**, you can now add rows to the spreadsheet for new coverables. If you exported **All from this version**, you can perform any of the following operations:

- Add new buildings or locations
- Make changes to existing buildings or locations
- Delete existing buildings or locations

## Using the Spreadsheets

You can send exported spreadsheets to agents or customers who then use any compatible spreadsheet application to fill in the needed details, adding one row per coverable. After you receive a filled-in spreadsheet, you can open it in your spreadsheet application to visually check the information. Then you can import the spreadsheet into PolicyCenter. Unless you are importing a template spreadsheet, import the spreadsheet into the same policy transaction (job) from which it was exported or a policy transaction (`job.basedOn`) the exported policy transaction. PolicyCenter displays an error message and aborts the import if the spreadsheet does not match the current policy transaction.

The exported spreadsheets have the following characteristics:

- The individual worksheets of the spreadsheet are protected and designed to be used while protected. If special use cases require a worksheet to be unprotected, the protection password is 1234. This password can be changed as explained in “Changing the Spreadsheet Protection Password” on page 483 in the *Configuration Guide*.
- Some columns of the spreadsheet are read-only so that you do not change their contents. These columns correspond to read-only fields in the PolicyCenter database. Read-only cells have a light gray background and an italic font.

- Capitalization of data in the spreadsheet is significant. For example, you attempt to add two buildings to the same location, but spell the city name with different capitalization in each of the two rows. The import operation displays an error because it assumes you are attempting to change the city name during an Add operation. Only the first building is added if you complete the import operation.
- Some columns contain lists that have been populated with the contents of a PolicyCenter typelist. In these columns, you must either select a value from the list or type the value exactly as it is listed, including capitalization.
- Numbers are exported as text to preserve their formatting. For example, class codes must be preserved with leading zeros, because 0034 is not equivalent to 34. Numbers are converted back to their appropriate types during the import operation.
- Policy data import ignores columns with headings that it does not recognize. Therefore, you can add notes or other information to the spreadsheet in unused columns, provided the column heading does not conflict with column headings used by policy data import/export. Any data you enter in such columns is not imported.
- You can import a spreadsheet only when the imported spreadsheet is basedOn the exported spreadsheet. You cannot import a spreadsheet that was exported from a different policy or from a different policy transaction in the same policy. To help you match spreadsheets to policies and transaction numbers, PolicyCenter suggests file names when saving exported spreadsheets. The pattern of the file name varies depending on the type of spreadsheet.
  - Template spreadsheets can be imported into any policy or policy transaction and can be used only for adding new locations or buildings. Template spreadsheets use the following file name format:  
`Template_FriendlyEntityName_yyyymmdd-hhmm.xlsx`
  - Non-template spreadsheets must be imported into the same policy and policy transaction from which they were exported. For this reason, non-template spreadsheets use the following file name format:  
`Policy_PolicyNumber_JobType_JobNumber_FriendlyEntityName_yyyymmdd-hhmm.xlsx`

`Policy_PolicyNumber` appears at the beginning of the file name only when the policy transaction is not a submission policy transaction. PolicyCenter does not assign a policy number until a submission policy transaction is quoted and bound.

### Examples of Export File Names

The following list contains example filenames of exported spreadsheets:

- `Template_Commercial_Property_Locations_20120514_1432.xlsx`
- `Template_Commercial_Property_Buildings_20120516_0954.xlsx`
- `Submission_16004467_Commerical_Property_Buildings_20120601_1014.xlsx`
- `Policy_5246715349_Policy_Change_16185124_Commercial_Property_Buildings_20120527_1602.xlsx`

The string portions of file names are exported in the selected language and are provided only as a convenience to users. PolicyCenter does not use the file name to match export policy transactions to import policy transactions. Instead it uses hidden data in the spreadsheet to match the policy transactions. If the policy transactions are not an exact match but can be linked through the `basedOn` property, PolicyCenter displays a message that the policy transactions do not exactly match. If the export and import policy transactions do not match, PolicyCenter displays an error message and does not complete the import operation.

Use care when choosing localized names containing prohibited characters in such items as file names and spreadsheets, or that are potentially dangerous when evaluated by operating systems. For example, Microsoft Windows (en\_US) prohibits the use of the following characters in file names:

`\ / : * ? " < > |`

PolicyCenter converts spaces in the file names to the underscore characters.

### Adding Buildings and Locations

When using the Commercial Property Buildings spreadsheet to add buildings, you can do any of the following:

- Add buildings to existing locations
- Add buildings to new locations
- Add multiple buildings to the same new location

In the Commercial Property line, buildings cannot exist on their own—they are child entities of locations. When a new building is created, it must be attached to a location, either an existing or newly-created location. In the base configuration, importing a building that has the **LocationID** of an existing location creates the new building in the existing location.

Importing a building that has a blank or unused **LocationID** creates the new building in a new location.

Importing multiple buildings with identical invalid **LocationID** fields creates a new location containing all of the new buildings. PolicyCenter converts the invalid **Loc ID** columns from the spreadsheet into valid **LocationID** field values when the imported spreadsheet is accepted.

**Note:** The **LocationID** field is key. If the **LocationID** does not exist in the PolicyCenter database, the import operation creates a new location, even if all of the address fields exactly match an existing location. To determine the **LocationID** of all of the locations defined within a policy, export a Commercial Property Locations spreadsheet and find the location IDs in the **Loc ID** column.

#### To add buildings to an existing location in the spreadsheet

1. In the spreadsheet editor, set the **Action** column to **Add**.
2. Leave the **Bldg ID** column blank, then fill in the other building data columns as needed. Stop when you reach the **Loc ID** column.
3. Switch to the worksheet tab named **unpopulated entity.CPLocation**. This worksheet contains a row for each location that has no buildings.
4. Locate the row representing the location to which you want to add the building. Select the individual cells that contain all of the location data and copy them to the clipboard.  
Because the worksheets are protected, you cannot copy and paste entire rows. Therefore, you must select and copy only the needed set of individual cells.
5. Switch back to the **Commercial Property Buildings** worksheet tab and select the **Loc ID** cell of the building you are adding.
6. Paste the clipboard contents to fill in the remaining cells with the exact data that defines the location.
7. Repeat these steps for additional buildings.
8. Save and import the spreadsheet.

#### To add buildings to new locations in the spreadsheet

1. In the spreadsheet editor, set the **Action** column to **Add**.
2. Leave the **Bldg ID** column blank, then fill in the other building data columns as needed, until you reach the **Loc ID** column.
3. Leave the **Loc ID** column blank to create a new location for the building.
4. Fill in the remaining location data columns as needed. Note that to successfully import the location, you must supply a territory code in the **Terr Code** column that is valid for the specified location. The spreadsheet does not provide a list of territory codes.

5. Repeat this steps to add more buildings, each at a new location. Upon import, PolicyCenter creates a new location each time it imports a building that has a blank Loc ID field.
6. Import the spreadsheet.

#### To add multiple buildings to the same new location in the spreadsheet

1. In the spreadsheet editor, set the Action column to Add.
2. Leave the Bldg ID column blank, then fill in the other building data columns as needed, until you reach the Loc ID column.
3. Type any non-blank value in the Loc ID column. For example, type “HomeOffice”.
4. Fill in the remaining location data columns as needed.
5. For additional buildings at the same location, repeat steps 1 and 2. Then copy the remaining range of cells beginning with Loc ID, to the corresponding range in each subsequent row. Note that each building with an identical Loc ID value is added to the same location.
6. Save and import the spreadsheet.

### Handling Policy Changes

You can make changes to buildings and locations in either a submission or change policy transaction. In a submission policy transaction, you can change or delete buildings and location that have been previously entered or imported but not yet bound or issued. In a renewal policy transaction, you can add, change, or delete existing buildings or locations.

#### To make changes to buildings or locations

1. In a PolicyCenter submission or change policy transaction, export a buildings or locations spreadsheet. PolicyCenter does not allow you to change location information in a buildings spreadsheet. PolicyCenter displays an error upon importing the spreadsheet. Change only building information in a buildings spreadsheet; change location information in a locations spreadsheet.
2. In the spreadsheet editor, select the appropriate action in the Action column for the coverable:
  - Blank – Change building or location data. Make the necessary changes within the spreadsheet row.
  - Add – Add new buildings or locations. Make the appropriate additions by following the same steps as described in the previous section.
  - Delete – Remove buildings or locations. No changes to the building or location data are necessary.
3. Repeat step 2 for each building or location that must be changed.
4. Save and import the spreadsheet to PolicyCenter.

### Importing a Spreadsheet

This topic provides step-by-step instructions for importing a spreadsheet to a PolicyCenter policy transaction.

#### To import a spreadsheet into PolicyCenter

1. In PolicyCenter open the policy transaction that matches the exported spreadsheet. You can determine which policy transaction to open by examining the file name of the exported spreadsheet. File naming conventions are described in “Using the Spreadsheets” on page 466.
2. Go to the Buildings and Locations screen, then click **Spreadsheet → Import**.

The **Import** option only appears if the policy transaction is in edit mode. If the **Import** option is not present when you click **Spreadsheet**, make sure the policy transaction is in edit mode by clicking **Edit Policy Transaction**.

**IMPORTANT** A large import operation can overwhelm the PolicyCenter server by creating a very large bundle file. The exact number of spreadsheet rows that can be imported without problems depends on system configuration, available memory, system load, and other factors. When the limit is reached, the PolicyCenter application server stops responding. To ensure satisfactory import performance, Guidewire recommends limiting the number of rows imported in a single operation to 1000. To import more rows, you can split the spreadsheet into multiple spreadsheets by cutting and pasting rows. You can cut data without un-protecting the spreadsheet by selecting the range of cells rather than selecting entire rows.

3. In the **Import From Spreadsheet** screen, click **Import**, then navigate to and select the spreadsheet to import. Click **Import** to proceed with the import operation.

On import, PolicyCenter detects the language of the spreadsheet. The spreadsheet can be in any language supported by the PolicyCenter instance, without regard to the language preference of the user importing the data.

For example, PolicyCenter is configured for English, French, and German. In PolicyCenter, an English-speaking user exports a spreadsheet in French. The French-speaking insured edits the spreadsheet and returns it to the carrier. A German-speaking user then imports the spreadsheet into PolicyCenter.

4. After the import operation completes, review the **Import From Spreadsheet** screen to view the results. At this point, the import operation is not complete and can be abandoned if needed. You can now do any of the following:

- View the **Import Summary** to assess the quality of the import operation. You can view the number of locations or buildings read, edited, added, and removed, and the number of rows that had errors. Any rows with errors are not imported.
- Click **Show Changes** to view a comparison that shows the changes that will be made if you accept the changes and complete the import operation.

When importing a spreadsheet that contains a large number of changes, using **Show Changes** can potentially take a long time.

After you have exported a policy transaction, it is possible that a user can make preemptions that affect some of the exported data. When the spreadsheet is later imported, PolicyCenter handles such preemptions in the same way as if the preemptions occurred during data entry in PolicyCenter. For information about how PolicyCenter handles preemptions, see “**Preempted Jobs**” on page 503 in the *Application Guide*.

- Click **Save Log** to save a log file containing the import summary and error information.
- Click **Cancel** to cancel the entire import operation. If your import operation caused errors, consider whether to resolve these interactively in PolicyCenter or cancel and repeat the import operation after making the changes in the spreadsheet.

5. Click **Accept Changes** to complete the import operation and update the policy with the imported changes, additions, and deletions.

# Product Model Overview

This topic presents an overview of the Guidewire PolicyCenter *product model*, which defines how PolicyCenter presents products in the user interface.

This topic includes:

- “Product Model Representation” on page 471
- “Products Overview” on page 472
- “PolicyLine Pattern Overview” on page 472
- “Coverage Pattern Overview” on page 472
- “CoverageTerm Pattern Overview” on page 474
- “Availability Overview” on page 475
- “Understanding Offerings” on page 477

**See also**

- “PolicyCenter Product Model” on page 11 in the *Product Model Guide* for detailed information on how to configure the product model.

## Product Model Representation

The PolicyCenter product model provides the definitions of the products that PolicyCenter offers. These definitions are called *patterns*. It is the pattern that is responsible for creating the actual instances of a product, a policy line, or a coverage, for example.

PolicyCenter uses these patterns during the submission process to generate instances of policies or the subcomponents of policies. Most of the product model patterns have *pattern* in their name, the exception is the Product entity. The following sections describe the most important product model patterns:

- Products Overview
- PolicyLine Pattern Overview
- Coverage Pattern Overview

- CoverageTerm Pattern Overview

In addition to these main template patterns, there are also form patterns and modifier patterns. For more information, see “Configuring the Product Model” on page 13 in the *Product Model Guide*.

## Products Overview

A **Product** represents a type of policy available to a customer. Each product is a separate row item on the initial Guidewire PolicyCenter **Submission** screen. A product can be *mono-line*, having only one associated **PolicyLinePattern** (for example, the Workers’ Comp Product). Or, a product can be *multi-line*, having more than one associated **PolicyLinePattern**. For example, the multi-line Commercial Package product contains the General Liability, Commercial Property, and Inland Marine policy lines.

The following are important parts of the product definition:

- The name and description.
- The array of policy line patterns that represent the policy lines (lines of business) associated with this product. A mono-line product has one associated **PolicyLinePattern**, whereas a multi-line product has many.
- The array of question set patterns.

**See also**

- “Working with Products” on page 15 in the *Product Model Guide* for a description of the **Product** screen interface

## PolicyLine Pattern Overview

A **PolicyLinePattern** holds all the configuration information for a line of business policy line. It holds information on all the coverages and similar items of the **PolicyLine** in the form of coverage patterns, form patterns, modifier patterns, and so on.

The following are important parts of the **PolicyLinePattern** definition:

- The name and description.
- The coverages, exclusions, and conditions associated with the line of business.

**See also**

- “Working with Policy Lines” on page 23 in the *Product Model Guide* for a description of the **Policy Line** screen interface

## Coverage Pattern Overview

A *coverage pattern* holds all the configuration information for a particular type of Coverage associated with a Policy Line. For example, the `WCEmpLiabCov` coverage pattern holds the configuration information associated with the Workers’ Compensation Employers Liability coverage.

The following are important parts of a coverage pattern definition:

- The name and description
- The coverage category which this coverage is associated with
- The coverable, the object that this coverage covers
- The terms of the coverage captured through the *CoverageTerm Pattern Overview* associated with the coverage pattern

- The existence of the coverage pattern on the policy line pattern. Existence can be required, suggested, or electable.
- The availability of the coverage pattern. The availability is based on a combination of lookups, a Gosu expression, grandfathering, and offerings.
- The `Initialization script` Gosu script, which runs when a user selects a coverage through the PolicyCenter interface
- The `Removal script` Gosu script, which runs when a user deselects a coverage through the PolicyCenter interface

#### See also

- “Adding Coverages to a Policy Line” on page 24 in the *Product Model Guide*

## Understanding Categories

A *category* is a group of coverage patterns, exclusion patterns, and/or condition patterns. A category bundles one or more patterns into a group for purposes of grouping them in the PolicyCenter user interface. You can also search for electable coverages by their category.

An example of a category is the Condo Unit Owner category in the businessowners policy line, which contains coverage patterns for:

- Miscellaneous property
- Loss assessment
- Condominium owner limit
- Miscellaneous property deductible

## Coverages, Exclusions, Conditions, and Coverables Overview

A *coverable* is an exposure to risk that can be protected by the policy. A coverable might be any of the following:

- A tangible property item
- A location
- A jurisdiction
- The policy itself

Within PolicyCenter, Guidewire makes the policy line a coverable to represent the named insureds. Coverages, exclusions, and conditions are attached only to coverables. You can further subdivide coverables into *property* coverables and *liability* coverables.

- *Property* coverables are things with physical attributes (height, weight, value, construction type, age, and similar attributes, for example).
- *Liability* coverables are operations represented typically by class codes (coal mining, personal auto operation, for example).

### Coverages

In contrast, a *coverage* is protection from a specific risk. Coverages are always attached to a coverable. You can divide coverages into the same two types as well: property and liability. For example, on an auto policy, a collision property coverage protects the vehicle owned by the insured. A liability coverage protects the driver for damage done to a vehicle owned by someone else. Liability coverage provides insurance for the operation of the vehicle. It does not provide insurance for the car, bus, or snowmobile.

Using a vehicle as an example:

- **Theft of items in a car** – The coverable is the vehicle and the type of loss is theft.

- **Car collision** – With collision coverage, the coverable is the vehicle owned by the insured. With comprehensive coverage, the coverable is the whole policy, covering damage to the other vehicle through liability.

## Exclusions and Conditions

Similar to coverages, exclusions and conditions are always attached to a coverable. While a coverage defines protection from a specific risk, an exclusion defines a risk that is explicitly not protected. Conditions define other contractual obligations of the insurance policy that are neither a coverage nor an exclusion.

Using a vehicle as an example again:

- **Loss while under the influence of alcohol** – The coverable is the whole policy, meaning the insured. The exclusion exempts any loss while the insured is under the influence of alcohol.
- **Policy-wide deductible** – The coverable is the whole policy, meaning the insured. The condition stipulates that all coverages use the same deductible.

## Existence

Simply because a coverage, exclusion, or condition is available to a policy does not imply that the coverage has been selected by a user. PolicyCenter specifies a coverage, exclusion, or condition on a Policy in one of the following ways:

Required	Coverage, exclusion, or condition that is on the policy (selected) and that a user cannot remove
Suggested	Coverage, exclusion, or condition that is on the policy (selected) by default, but the user can remove (unselect) it, if desired.
Electable	Coverage, exclusion, or condition that is not on the policy (unselected), and the user can select it, if desired.

All of these combinations are, of course, subject to the Availability of the Coverage. If a coverage, exclusion, or condition is not available, it simply does not appear on the policy.

## CoverageTerm Pattern Overview

A CoverageTermPattern holds the configuration information for the *terms* of a particular coverage. (Equivalent patterns exist for exclusions and conditions.) A coverage term specifies the extent or degree of coverage, or specifies an attribute of coverage. Terms are usually limits or deductibles. They can also specify scope of coverage, such as whether coverage form is Basic or Broad. Terms can specify a selection or an exclusion that is specific to a particular coverage. For example, is air conditioning failure covered as part of Boiler and Machinery coverage?

PolicyCenter defines a coverage term pattern by its *type* and its *model type* (among other things).

Type	Specifies how the user selects the value of the coverage term. For example, do you choose from a drop-down list, enter a numeric value, or select from a predefined set of packaged values (100/200/300)?  You set the coverage term type in the New Coverage Term dialog while you create the coverage term.
Model type	Specifies what the value measures. For example, is this a limit or a deductible? Systems integrated with PolicyCenter can use this information to correctly interpret the coverage term pattern information.  You set the coverage term model type in the Basics tab, after you create the coverage term.

While there are some differences between the different CoverageTermPatterns, they do share some common attributes. The following are important parts of the CoverageTermPattern definition:

- The Name and Description.
- The database table column to use for the coverage term.

- The Priority of the coverage term. This priority affects the order in which PolicyCenter renders it in the user interface. PolicyCenter renders lower numbers first.
- The Default Value text field, which can be used to set the default value of the coverage term.
- The Model Type of the coverage term. That is, if the `CoverageTermPattern` represents a Limit, a Deductible or something else, such as an election.

**See also**

- “Defining Coverage Terms” on page 32 in the *Product Model Guide*

## Availability Overview

PolicyCenter allows you to specify whether or not a coverage or other pattern is available based on a variety of factors. These factors include the start and end effective dates, industry code, underwriting company, and policy transaction type. You can also write a script to determine availability based upon various factors, including answers to question sets.

Availability is also determined by the following:

- |                                                                                                               |                                                                                                             |
|---------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Reference Date</li><li>• Grandfathering</li><li>• Offerings</li></ul> | <ul style="list-style-type: none"><li>• Availability lookup tables</li><li>• Availability scripts</li></ul> |
|---------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|

Typically, availability is determined by the insured jurisdiction, underwriting company, and the reference date. You can configure these, and additional dimensions, in availability lookup tables.

**See also**

- “Configuring Availability” on page 83 in the *Product Model Guide*
- “Checking Product Model Availability” on page 105 in the *Product Model Guide*

## Determining the Reference Date

Availability is frequently determined by start and end effective dates. The *reference date* is compared against the start and end effective dates in the availability lookup tables to determine whether the pattern is available. This topic describes why patterns change over time and how the application determines the appropriate reference date to use.

The coverages, exclusions, and other aspects of a policy’s definition that can be used by an insurance carrier change over time, as do the rates that are applied to these. These changes occur because carriers typically file their rates, forms, and underwriting rules with the regulatory body for each jurisdiction, and these filings change over time. (Examples of regulatory bodies are the state departments of insurance in the United States.) For example, a carrier might start offering a new coverage, along with its form, rates, and the underwriting rules. This new coverage becomes available on a particular date and is not be available retroactively. PolicyCenter tracks these dates through effective and expiration dates in the availability tables of the product model. It is the responsibility of the rating engine to maintain effective and expiration dates on the rate tables.

The reference date that is compared against the effective and expiration dates of an availability or rating table is not always the same. It might be the start date of the policy period, the current date, or another date entirely. To handle this, PolicyCenter has a framework to determine the appropriate reference date before determining the availability of patterns in the product model.

The first step to determining the appropriate reference date is to determine the type of reference date. The reference date types are:

- **Written Date** – The date something was created or processing was started.
- **Effective Date** – The date something was applied to a policy.
- **Rating Period Date** – For Workers' Compensation only, this date is based upon the anniversary date of the policy.

**See also**

- “Setting the Reference Date” on page 90 in the *Product Designer Guide*

## Making a Pattern Available by Policy Transaction Type

You can specify whether to use a pattern based on the policy transaction type. Carriers may want to make a coverage or other pattern available to new business before allowing for renewal business. Carriers do this to make changes available as soon as possible without having to update in-process renewals.

For example, a pattern can be available for all new business, of any policy transaction type except renewals, starting on a particular date. Specify a later start effective date for renewals. In this way, the pattern is made available sooner for most usage, including submissions and policy changes, but later for renewals.

**See also**

- “Defining a Pattern as Available on a Job-by-Job Basis” on page 87 in the *Product Model Guide*

## Grandfathering Overview

Grandfathering allows you to continue to offer a coverage or other pattern to existing customers, even though the pattern is not otherwise available. Therefore, you cannot use the pattern with new customers or as an addition to policies of existing customers. However, the pattern is not automatically removed from existing usages.

Grandfathering is available on:

- Coverages
- Exclusions
- Conditions
- Coverage terms
- Coverage term options
- Coverage term packages
- Modifiers
- Offerings

Grandfathering is determined by the jurisdiction, underwriting company, and end effective date. Once configured, grandfathering is automatically applied, typically during renewal policy transactions. Grandfathering can also be applied during renewal upon conversion from a legacy system.

For example, the carrier decides to offer a particular coverage in California until December 2007, then grandfathers the coverage until December 2009. The carrier also offers this coverage in Nevada until October 2007, then grandfathers the coverage until May 2010.

**See also**

- “Defining Grandfathering” on page 88 in the *Product Model Guide*

## Reloading Availability Overview

You can make changes to availability data and upload these changes to a running PolicyCenter server or clustered group of servers. The types of availability data you can upload are:

- Lookup tables
- Availability scripts
- Grandfather states

### See also

- “Reloading Availability Data” on page 95 in the *Product Model Guide*
- “Reloading Availability Example” on page 96 in the *Product Model Guide*

## Understanding Offerings

Some insurers offer variations of their policies by customer or how the sale is being made. Offerings let you define different product types for different types of buyers. You can use offerings for the following use cases:

- **Business-specific products** – An insurer offers a business program that consists of a set of common coverages. The insurer offers specialized products based on this business program. These specialized products are offered to retailers, auto shops, and the hospitality industry, among others. These specialized products offer coverage levels appropriate for each business type.
- **Affinity groups** – Some insurers write policies that are based on a group membership of the insured. These affinity group policies offer a subset of the available coverages, group-specific default values, and possibly group-specific value choices for the coverage terms. Often the policy is subject to a special rate agreement, and so must obey various restrictions on the coverages and terms offered.
- **Programs or tiers of coverages** – These are similar in concept to the affinity group. The classic example is Bronze, Silver, and Gold programs. The customer can choose increasing levels of coverage at increasing cost. Programs can also offer different coverages.

A carrier may have hundreds of types of offerings. Insurers want to be able to create these offerings quickly, often based on a similar pre-existing product. PolicyCenter provides the tools to quickly and easily create offerings based on an existing product definition. You start with the base product definition, and then simply tailor it to define your specific offering.

If the product contains offerings, you can select an offering in the submission, issuance, policy change, renewal, and rewrite policy transactions.

This topic contains the following:

- “Filtering the Product Model” on page 477
- “Offering Question Sets” on page 478
- “Working with Offerings” on page 478

## Filtering the Product Model

The selected offering is the last check in product model availability. If all other checks say that a coverage or other pattern is available, then the offering is the final check that can set it to unavailable. If the other checks say that the coverage is unavailable, then the offering cannot make it available.

Offerings can filter the following parts of the product model:

- Policy terms
- Policy lines: you can filter out policy lines in a *Package Policy*.
- Coverages

- Exclusions
- Conditions
- Coverage terms
- Coverage term options and packages
- Modifiers
- Question sets

## Offering Question Sets

The **Offerings** screen allows you to choose an offering. The **Offerings** screen displays any question sets included on the product that have their type set to **Offering Selection**. Answers to questions on this screen can filter the offerings available from the **Offering Selection** dropdown menu.

### See also

- “Offerings and Question Sets” on page 102 in the *Product Model Guide*

## Working with Offerings

The following steps guide you through selecting an offering and changing the offering.

**Note:** The instructions assume that you are familiar with creating a submission. For complete instructions on how to create a submission, see “Creating a Submission” on page 87.

**1.** Create a submission for a **Businessowners** policy.

**2.** Continue to the **Offerings** screen.

The **Silver** and **Platinum** offerings are available if the questions have default answers and there are no other triggers filtering the selections.

**3.** Select **Yes** to the question **Is the customer a member of Partners Alliance?** The **Partners** offering is added to **Offering Selection** drop-down menu.

**4.** Select **Partners** from the **Offering Selection** drop-down menu.

**5.** Click **Next**, and continue to the **BusinessOwners** screen. Because you selected the **Partners** offering, the **Policywide Property Deductible** is optional (and the check box is initially blank). This coverage is required if no offering is selected. Offerings can change what appears on this page, and on other pages in the wizard.

You can go back and change the offering.

**6.** Click **Offerings** in the left sidebar.

**7.** From the **Offerings Selection** drop-down menu, choose **<none selected>**.

**8.** Click **BusinessOwners** in the left side bar to return to that screen. Notice that the **Policywide Property Deductible** is now required (it cannot be unselected). PolicyCenter displays a validation message at the bottom of the screen. The message says, “Policywide Property Deductible is required and has been added.”

# Policy Revisioning

This topic describes the way that PolicyCenter stores information about an insurance policy as it undergoes changes over time.

**Note:** In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

This topic includes:

- “What Is a Policy Revision?” on page 479
- “Structure of Revisioning Across Effective Time” on page 487
- “Out-of-sequence Jobs” on page 501
- “Preempted Jobs” on page 503
- “Applying Changes to Future Renewals” on page 507
- “Revisioning Rewrite Jobs” on page 507
- “Summary of Revisioning Terminology” on page 508
- “Revisioning Properties Reference” on page 509
- “Details of Merging and Applying Changes” on page 512
- “Policy Differences Between Revisions” on page 516

## What Is a Policy Revision?

An insurance policy may change one or more times during its lifetime within PolicyCenter.

Policies may change in the middle of a period due to:

- Adding a driver to a policy
- Changing coverage amounts
- Adding a vehicle

- Canceling the policy
- Reinstating the policy

The complete history of all policy changes in legally-binding policies must be carefully tracked, not merely stored in the latest version of the policy. The policy history might be needed for legal auditing, customer service, financial reports, or tracking how much to charge customers for a change.

PolicyCenter stores the policy history as a series of *policy revisions*. Policy revisions are like snapshots of the policy on date the revision was bound (when it became legally binding). When a revision is bound, that revision represents the legally-enforced truth of that policy for all effective dates within a single policy period. However, PolicyCenter preserves older versions of the truth for that policy as historical records. Both the enforced versions and the historical versions persist and can be used or compared as needed.

In PolicyCenter, policy revisions are often referred to as *branches*. You can think of a branch as a graph of objects with *PolicyPeriod* at the root. The branch collectively represents a policy for one contractual period as of one moment in real-world time.

To track policy changes over time, a policy must be considered in two different time dimensions:

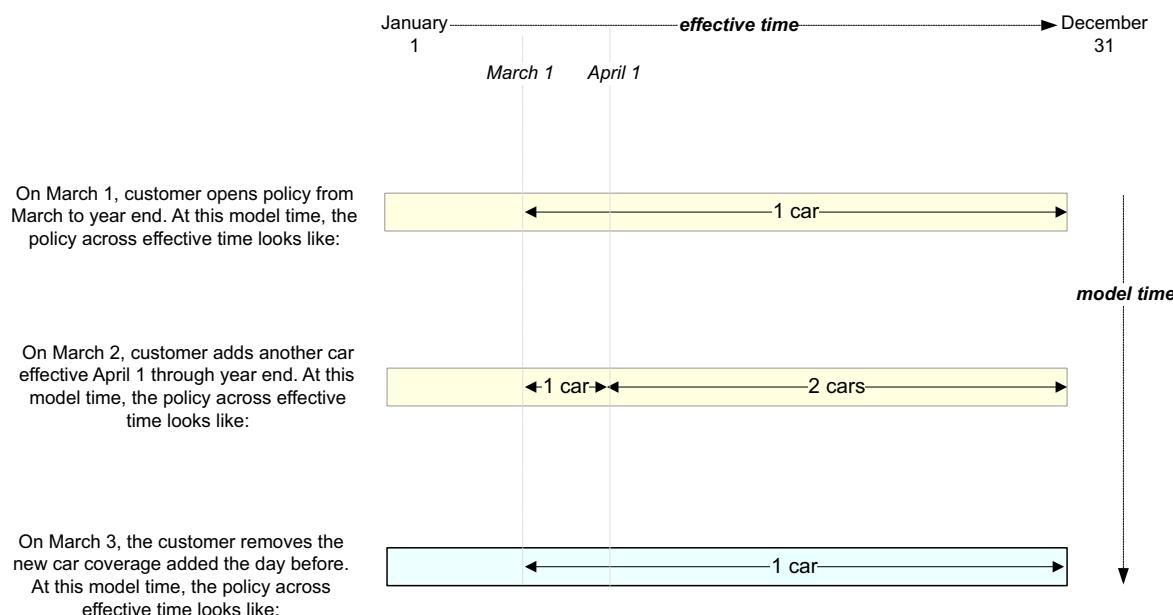
Dimension	Definition	How PolicyCenter uses this dimension	Examples
model time	The actual real-world time when policies are created or jobs (policy transactions) are bound. This is like tracking the history of previous changes in any online system that has an audit trail.	When a branch is bound, PolicyCenter sets its branch <i>model date</i> to match the real-world date it was bound. Additionally, PolicyCenter increments the policy revision's <i>model number</i> , which is an integer value that indicates the relative order of multiple versions of the same contractual policy revision. The bound revision with the latest model number is always the currently-active legally-enforced policy revision for that effective time range. Changes that happen later supersede earlier versions of the policy for the policy period's effective time range. However, PolicyCenter keeps older branches in the database. Older branches are required to view the policy history. Use this to generate reports of the legally-binding state of the policy at a model date earlier than today.	<ul style="list-style-type: none"> <li>• Suppose a customer takes out a policy on a red car. Later the customer calls and apologizes and says it is a blue car. Yet again the customer calls and says the car really is green. The policy covers the same policy period with the same effective time but there are three different model dates with changes.</li> <li>• Policy reports run every quarter must reflect the state of policies on the last day of each quarter, which is the model date. However, you may not run the report until several weeks later. It is important that the system can query the policy as of a specific model date and ignore all changes made later (in actual time) after the model date.</li> </ul>
effective time	The time dimension of the policy itself within the policy period. For example, what time range does the policy cover? This dimension of time is unique to a policy system.	If the policy period is one year, each PolicyCenter policy period records the policy information for one year of effective time. Some objects on the policy may only exist for some range of effective time, or have different property values for different ranges of effective time.	<ul style="list-style-type: none"> <li>• Suppose a policy covers one car for an entire calendar year. Halfway through the year, the insured individual buys a used car. The policy covers the first car for the whole year of effective time. The policy covers the added car for the second half of effective time.</li> <li>• A change to an effective date in the future: a customer calls to say they will add a car to the policy as of the beginning of the next month.</li> <li>• A change to an effective date in the past: canceling a policy effective last month.</li> </ul>

To contrast the two dimensions of time, suppose a customer calls on March 1. The customer wants to add a car to the policy as of the beginning of the next month, April 1st:

<b>Model date:</b>	March 1
<b>Effective date:</b>	April 1, effective until the end of the period

The following diagram shows a policy across effective time and model time.

### Comparing Effective Time and Model Time



#### Key

**Enforced PolicyPeriod**

After multiple changes, only the most recently updated PolicyPeriod is legally enforced. PolicyCenter keeps older versions for historical reasons, such as reports with model date earlier than today.

**Historical PolicyPeriod**

**IMPORTANT** Be sure you understand the differences between *effective time* and *model time* before proceeding through this topic. These concepts are extremely critical for understanding the complex sequencing issues discussed later.

## Basic Revisioning Structure of a Policy

Think of a policy as a container of *contractual periods*, each with a specific range of effective time. A contractual period is a single policy term from the date the policy goes into effect (the effective date) to the date it expires (the expiration date). For example, if a homeowners policy has a year long period, calendar year 2008 is one

contractual period. If that 2008 policy is modified, then PolicyCenter saves each version. Each version represents the same policy in the same period.

The period can be any length and start on any day of the year. A period can be six months, three months, or any arbitrary length. The main rule is that one policy cannot have contractual periods that overlap in effective time, not including sections of contractual periods where a policy was canceled or rewritten.

**Note:** The contractual periods might overlap if there was a cancellation and a new contractual period was created after the cancellation date. However, their effective time (the non-canceled time range) for the periods must never overlap.

When a policy renews, the renewal job (policy transaction) creates another contractual period with a different range of effective time. For example, a renewal job for a 2008 policy period would clone the data for that policy for a new contractual period with different effective dates for 2009.

Each one of these snapshots of the policy for one period is a *branch*. PolicyCenter represents each branch in the contractual period as a `PolicyPeriod` entity instance. This entity instance is a container for the rest of the objects on the policy, for example vehicles, coverages, and policy contacts. PolicyCenter assigns each period a unique period ID, which is stored in a `PolicyPeriod` in its `PeriodID` property. That value identifies and links all branches for that contractual period.

As part of making a branch legally enforced, PolicyCenter performs a process called binding. This process is also called binding a branch or promoting a branch. The result is a promoted branch or a bound branch. When binding a branch, PolicyCenter sets the `ModelDate` property in the `PolicyPeriod` to the real-world date it was bound. Additionally, if there are earlier versions of this `PolicyPeriod` entity instance in the contractual period, PolicyCenter increments the `PolicyPeriod` model number (the `ModelNumber` property). It sets it to one greater number than the most recently bound earlier revision in this contractual period. These model time properties let PolicyCenter track what is the legally-enforced version of the policy for that period.

Each `PolicyPeriod` also includes a `MostRecentModel` property that is `true` if this `PolicyPeriod` is the most recently bound branch for this contractual period. When a branch is bound, if there was another branch in that contractual period, PolicyCenter sets two things. First PolicyCenter sets this property to `false` on the previous branch and sets it to `true` on the newest branch in the same database transaction. Technically, this is redundant with checking for the highest model number (`ModelNumber`) for all `PolicyPeriod` entities in this contractual period (those that share the same `PeriodID`). However, this property is provided to simplify queries that work only with the latest bound branch in any given period. The `MostRecentModel` property is very useful for writing reporting queries.

If a `PolicyPeriod` cannot be modified because the branch is bound, withdrawn, or discarded, PolicyCenter sets its `Locked` property to `true`. This locking prevents accidental changing of that `PolicyPeriod` or any of its subobjects. PolicyCenter enforces this locking at the application level.

**Note:** You can customize application logic before promoting a branch. For more on this topic, see “Customizing Behavior Before Promoting a Branch” on page 231 in the *Integration Guide*.

### Subobjects

Every policy revision branch is represented by a `PolicyPeriod` entity instance at the root of a complex graph of subobjects such as policy lines, vehicles, coverages, and many others. The entire hierarchy of Guidewire entities are cloned in the database into new rows during policy changes, renewals, or other jobs that result in cloning everything in a branch. In contrast, a submission job’s branch is not cloned from another branch.

PolicyCenter must identify that some rows in the database represent the same real-world thing. This is true for the following cases:

- **Across model time** – PolicyCenter typically represents one object (such as a driver or a vehicle) more than once across model time. PolicyCenter creates one instance each time it copies the branch due to a policy change job or other job. To understand differences between two historical periods, PolicyCenter needs to know they represent the same object not multiple different objects.

- **Multiple periods** – One object (such as a driver or a vehicle) might exist in multiple contractual periods. To understand differences between two periods, PolicyCenter needs to know they represent the same object not multiple different objects.
- **Across effective time** – Some objects change across effective time within one branch. To understand that these are the same object across effective time, PolicyCenter must know these represent the same object, not different objects. For example, suppose you have a car on a policy and then need to change the license plate number. The database contains two rows for the car: one with the original license plate number, one with the new license plate number. This topic discussed further in “Structure of Revisioning Across Effective Time” on page 487.

For these reasons, PolicyCenter knows which rows represent the same object because they share an ID called a *fixed ID*, stored in its `FixedID` property. If the fixed IDs for two vehicles match, they are versions of the same vehicle, not two different vehicles. If the fixed IDs do not match, they represent different vehicles.

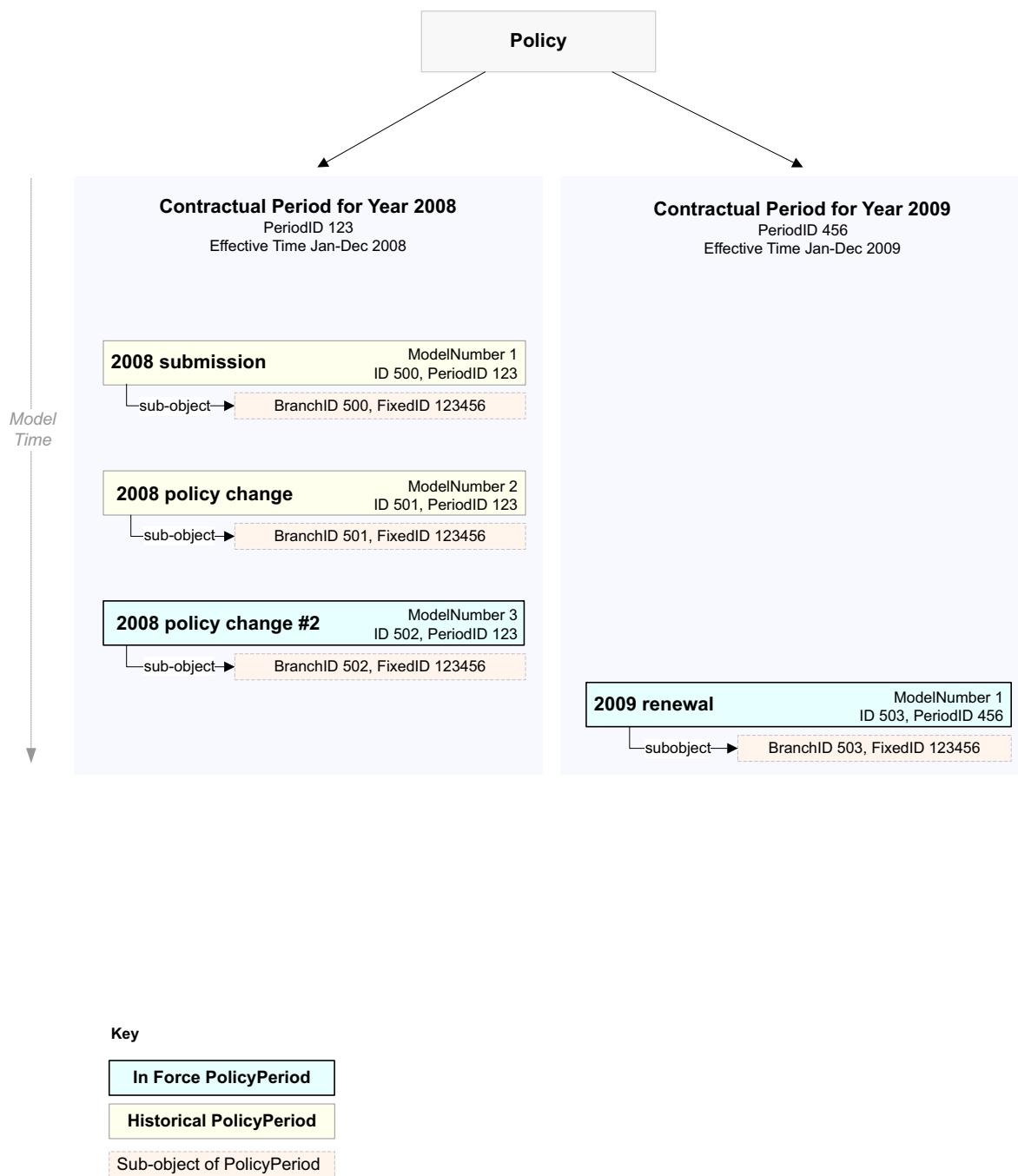
**Note:** In previous releases of PolicyCenter, the fixed ID was called a revision-independent ID (*RIID*).

Each subobject also contains a foreign key to the `PolicyPeriod` entity instance that contains it. This foreign key is called a *branch ID* and is stored in the subobject’s `BranchValue` property. This foreign key always matches the `PolicyPeriod` entity instance’s `Id` property. Remember that this foreign key references the `PolicyPeriod` unique `Id` property, not the `PeriodID` property that identifies related `PolicyPeriod` entities in one contractual period.

The following diagram shows the structural relationship of simple policy with two contractual periods. Note in the diagram:

- Within one contractual period, each `PolicyPeriod` entity instance shares the same period ID.
- Each `PolicyPeriod` entity instance has a model number that increments for each revision in the contractual period each time a change is made in model time (real-world time).
- Each subobject contains a branch ID that identifies its root `PolicyPeriod` entity instance, and it matches the `PolicyPeriod.Id` property
- Each subobject has the same fixed ID when the subobject exists in multiple branches and even across contractual periods. For example, a car’s data that was modified has the same fixed ID in each branch that references it. The fixed ID is also the same in renewal periods if that car is still covered in the renewal period.

## Revisioning Structure and IDs



## The Policy Period and Effective Date Fields

The `PolicyPeriod` entity is the root of the graph of revised objects. The `PolicyPeriod` is the branch that all `EffDated` objects in the graph point to. The `PolicyPeriod` defines the `PeriodStart` and `PeriodEnd`.

Although the `PolicyPeriod` entity is the root of the revised graph, it does not behave like other revised objects. The `PolicyPeriod` entity delegates to the `EffDatedBranch` interface, but not to `EffDated`. Because the `PolicyPeriod` is not `EffDated`, the `PolicyPeriod` is handled differently than everything else in the policy period graph and does not have behaviors like splitting on slice mode edit. Therefore, any property, such as one containing data, a typekey, or foreign key, placed directly on `PolicyPeriod` behaves differently than one placed on `EffDated` objects such as `PolicyLine` or `PolicyLocation`.

A property added directly to `PolicyPeriod` is for the full term. These properties always have the same value from `PeriodStart` to `PeriodEnd`. You can still edit a property value on `PolicyPeriod` in a policy change, but that value replaces the former full term value. The value is not effective as of the effective date of the policy change job.

For example, the `PolicyPeriod` has unrevised properties that apply to the full term. Some of these properties are:

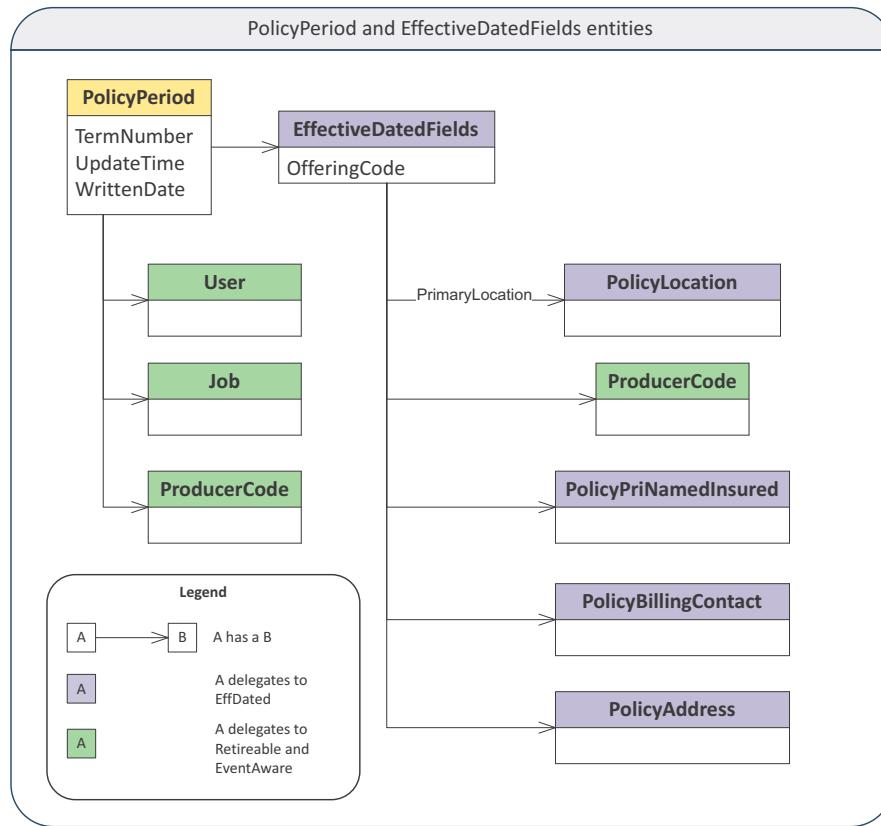
- `TermNumber` – An integer
- `CreateUser` and `UpdateUser` – Foreign key to `User`
- `Job` – Foreign key to `Job`

Revised properties related to the `PolicyPeriod` are off of the `EffectiveDatedFields` object. Some of these revised properties are:

- `OfferingCode` – A patterncode
- `ProducerCode` – Foreign key to `ProducerCode`
- `PrimaryLocation` – Foreign key to `PolicyLocation`
- `PrimaryNamedInsured` – Foreign key to `PolicyPriNamedInsured`
- `BillingContact` – Foreign key to `PolicyBillingContact`
- `PolicyAddress` – Foreign key to `PolicyAddress`

These properties are all items that are revised but not tied to a `PolicyLine`. For convenience, the `PolicyPeriod` object defines derived properties that enable you to access these revised properties. For

example, you can access `EffectiveDatedFields.BillingContact` through the derived property `PolicyPeriod.BillingContact`.



Arrays and one-to-one relationships of EffDated objects behave the same whether placed on the `PolicyPeriod` or an `EffDated` object. All array or one-to-one relationships have a foreign key back to the object to which they are attached. Arrays of EffDated objects attached to a `PolicyPeriod` vary across effective time. For example, the `PolicyPeriod` has an array of `PolicyContactRole` objects that are EffDated. You can determine array membership by the `EffectiveDate` and `ExpirationDate`. For example, on 1/1/2012, the array contains one `PolicyContactRole` object. On 3/1/2012, a policy change adds two `PolicyContactRole` objects. On 5/1/2012, another policy change removes the first `PolicyContactRole`.

### Deciding Whether to Add a Property to `PolicyPeriod` or `EffectiveDatedFields`

A property defined on the `PolicyPeriod` object behaves differently than a property defined on the `EffectiveDatedFields` object.

The `PolicyPeriod` has a `ProducerCodeOfRecord` foreign key which points to a `ProducerCode`. However, if you move the `ProducerCodeOfRecord` foreign key from `PolicyPeriod` to `EffectiveDatedFields`, there could be more than one `ProducerCodeOfRecord` for the period. The `ProducerCodeOfRecord` could vary over effective time.

Suppose you have a submission with the policy period extending from 1/1/2012 to 1/1/2013. The `ProducerCodeOfRecord` is Alpha:

- Foreign key on `PolicyPeriod` – The producer code of record for the submission is Alpha.
- Foreign key on `EffectiveDatedFields` – The producer code of record for the submission from 1/1/2012 to 1/1/2013 is Alpha.

On the submission, both ways of representing the producer code of record are effectively the same. Now you do a policy change effective 7/1/2012 and change the ProducerCodeOfRecord to Beta:

- Foreign key on PolicyPeriod – The producer code of record for the policy change is Beta. The producer code of record is effective for the full policy period.
- Foreign key on EffectiveDatedFields – The producer code of record is Alpha from 1/1/2012 to 7/1/2012, and then it is Beta from 7/1/2012 to 1/1/2013.

With the foreign key on an EffDated entity, you have a split in effective time when you change the policy period. The foreign key was one value, and at some date it changes to another value. When the foreign key is on the PolicyPeriod, any change is for the full period. For the policy change it is Beta for the full term—there is no concept that is Alpha for some dates and then Beta for other dates. You can still go back in history and see that the submission had a different value. This is the key difference between the two models.

## Structure of Revisioning Across Effective Time

Every policy revision (a PolicyPeriod entity instance) is the root of a complex graph of subobjects such as policy lines, vehicles, coverages, and many other objects. If one object such as a vehicle does not change across effective time, the database contains only one row for that object at that model time. However, if the object changes across effective time, the object is cloned into a new row in the database, differing in effective time begin and end dates. The insurance industry calls these begin and end dates the *effective date* and *expiration date*.

For example, suppose on March 1 a customer requests an auto policy for a red car, effective from March 1 through the end of the year. On August 1, the customer calls and says the car was painted today and is now blue.

PolicyCenter represents these changes across effective time for the vehicle as two rows:

- A red car with effective date March 1 and expiration date August 1 (typically at 12:00am).
- A blue car with effective date August 1 and expiration December 31

Note that the expiration date of the first row is the same as the effective date of the second row. This means the first row is effective up until the exact date and time of expiration, but not including that exact date and time. A row is effective at date specified\_date if the following equation is true:

```
effective_date <= specified_date < expiration_date
```

Effective dates are stored in the EffectiveDate property of each PolicyPeriod subobject. If the EffectiveDate property is null, implicitly the effective date of the subobject is the effective date of the PolicyPeriod that contains the object. The effective date of the PolicyPeriod object is in the PolicyPeriod.PeriodStart property.

Expiration dates are stored in the ExpirationDate property of each PolicyPeriod subobject. If the ExpirationDate property is null, implicitly the expiration date of the subobject is the expiration date of the PolicyPeriod that contains the object. The expiration date of the PolicyPeriod object is in the PolicyPeriod.PeriodEnd property.

The following diagram represents the structure of revisioning across effective time, showing a single vehicle subobject in an auto policy. Notice that a policy change of an existing object can sometimes split an object into two objects. Each object has different effective time ranges. These objects have the same FixedID values since they represent the same object. However, when adding an entirely new object such as a vehicle, the new entity instance has a different FixedID value. The new fixed ID shows that it represents a new vehicle, not a change to an existing vehicle.

## Sub-objects Across Effective Time

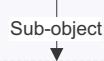
### Contractual Period

For the year 2008

New auto policy for a red car, effective for all days in 2008

#### PolicyPeriod for submission

ModelNumber 1



##### VEHICLE

Color: **RED**  
EffectiveDate: null  
ExpirationDate: null  
FixedID: 12345

A PolicyPeriod entity's sub-object has an EffectiveDate or ExpirationDate field to indicate the effective date range. If they are null, they implicitly use the start date and end date of the containing PolicyPeriod. In this case, both are null, which means the effective date range is the entire year.

Change the car's color to blue, effective August 1 through year end

#### PolicyPeriod for policy change

ModelNumber 2



##### VEHICLE

Color: **RED**  
EffectiveDate: null  
ExpirationDate: August 1, 2008  
FixedID: 12345

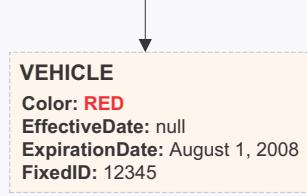


Policy changes can split entities. One has effective time range before the change's effective date, and another has changes after that date. Also note the same FixedID values means they are the same car, with different effective date ranges.

Add additional (new) car to policy, effective September 1 through year end

#### PolicyPeriod for policy change #2

ModelNumber 3



##### VEHICLE

Color: **BLUE**  
EffectiveDate: August 1, 2008  
ExpirationDate: null  
FixedID: 12345

##### VEHICLE

Color: **GREEN**  
EffectiveDate: Sept 1, 2008  
ExpirationDate: null  
FixedID: 67890

The different FixedID means it represents a different car.

### Key

Enforced PolicyPeriod

Historical PolicyPeriod

Sub-object of PolicyPeriod

## Unbound Policy Revisions

All legally-binding versions of a policy are kept as a series of ordered revisions in the database. However, PolicyCenter also stores revisions that are not part of the official legally-binding history of the policy.

Examples of unbound branches include:

- Branches for unfinished jobs, such as in-process policy changes. These are sometimes called *draft revisions* or *in-progress revisions*.
- Withdrawn policy changes, withdrawn by selecting **Withdraw Policy Transaction** before changes were bound.
- Rescinded cancellations (before changes were bound)
- Unbound versions (unselected versions) of a multi-version quoting job. At most one version can be bound for a job.

Non-bound revisions are stored in the same database tables as bound revisions, differing only in their **Promoted** property. If the **Promoted** property value is **true**, it was made legally binding. However, it may not be the only one in that period. The branch with the highest model number is the enforced branch for that contractual period, which is found most easily by using the Boolean property **MostRecentMode1**.

## Slice Mode and Window Mode Overview

There are two ways to access a branch's subobjects: *slice mode* and *window mode*.

In most cases, a job changes its policy data effective as of a specific date in effective time. This date is stored on a **PolicyPeriod** as the **EditEffectiveDate**. Most changes for a job (and sometimes all changes for that job) will happen in effective time as of that date. Because this is the typical way to edit a **PolicyPeriod** and its subobjects, PolicyCenter includes a special way to view the branch as of that effective date. PolicyCenter can hide subobjects that are not effective at the desired date. This way to view the branch is called *slice mode*, and is the most common way PolicyCenter accesses policy data. Most edits are made at this date, called *the slice date*.

When working with a **PolicyPeriod**, the **PolicyPeriod** is typically already in slice mode. For example, if you access a **PolicyPeriod** as a reference for a job, the PolicyCenter job has probably set the slice date to the job's effective date. This is true for almost all code that you will work with. You typically do not need to do anything special to setup the **PolicyPeriod** slice date if you want to work with the branch at the effective date. Simply access the properties on objects referenced from the **PolicyPeriod** in a job and work with it as a graph of objects. The slice date does not persist. PolicyCenter stores this only on the in-memory **PolicyPeriod** entity instance accessed by Gosu. PolicyCenter never stores this in the database row with the **PolicyPeriod**.

**Note:** The slice date (and more generally, the slice mode or window mode status) is stored on each in-memory copy of effective dated entities. This information is not persistent in the database row.

In contrast, sometimes you must view all versions of an object **across all dates in effective time**. This view is called *window mode*. For example, suppose a policy covered one car on an annual policy that starts at the beginning of the year. On March 1, the insured buys a new car and adds it to the policy. On March 10, she sells the old car and removes it from the policy.

- In slice mode viewed as of February 15, the policy contains only the first car.
- In slice mode viewed as of March 5, the policy contains two cars.
- In slice mode viewed as of April 1, the policy contains only the newer car.

**Note:** When you view a policy in slice mode at any given effective date, you might not see all cars that exist at some point in that policy period.

Instead, suppose you want to display a page that lists all cars that exist on the policy for any amount of time. For each car, you want to list the effective date ranges for each car. To do this effectively:

1. Get a list of all cars that were ever on the policy. There is one entry for each car, regardless of whether it changed over effective time.

2. For each car, iterate across all changes to that particular car across effective time. There is exactly one row in the database for each version of that car. For example, if the car has had three different colors, there are three rows in the database that represent that car. The set of three rows that represent this one car is called a *version list*. From the version list, you can access every version of the car in the period. In this case, the version list contains three versions of the car, each with a different color.

When you request all cars on the policy in slice mode, PolicyCenter automatically gets the correct version of each car as of the slice date. Also, PolicyCenter only returns the cars that are effective at the slice date.

In contrast, you ask for all the cars on the policy in window mode, there is no implicit slice date. PolicyCenter gets a version list for each car.

For more details about window mode, see, “Window Mode API Overview” on page 491.

## Slice Mode APIs

In most cases, you access a `PolicyPeriod` related to a job. In standard application contexts, PolicyCenter sets up the `PolicyPeriod` and sets the slice date to the effective date for the job. Typically you do not need to do anything special to set a slice date if you want to work with the branch at the effective date. Simply access the properties on entities referenced from the `PolicyPeriod` and work with it as a graph of objects.

To ensure you access a `PolicyPeriod` entity instance in slice mode, you can call its `getSlice` method:

```
// get a handle to the PolicyPeriod in slice mode with a specific date and SAVE the return value
slicedPolicyPeriod = aPolicyPeriod.getSlice(sliceDate)
```

The returned object represents the same `PolicyPeriod` entity instance, but it is a different object in local memory that Gosu specially marks as *in slice mode*. Remember to use the return value from the `getSlice`, not your original reference to the `PolicyPeriod`. The original in-memory copy of the entity instance is unchanged.

If you get properties on a slice mode object to access other objects, those objects are also automatically in slice mode. In typical code, you can navigate up or down the object graph hierarchy without worrying about the revisioning details. At any time you can get the `object.SliceDate` property to get the slice date.

In most cases, it is best to call `getSlice` on the root `PolicyPeriod` and navigate down the object graph from there. However, you can call `getSlice` on an individual revised subobject of `PolicyPeriod` if necessary.

For example:

```
autoLineExpiration = vehicle.getSlice(sliceDate).PolicyLine.PolicyPeriod.ExpirationDate
```

The foreign key links after the `getSlice` method implicitly use the slice date to find the right version of the `PolicyLine`.

**Note:** If a policy period is in slice mode and you get any subobjects, they are in slice mode automatically. It is redundant to call `getSlice` with the same slice date.

Be sure that the slice date that you pass to `getSlice` is in the effective date range for that object. If you try to pass a date outside the required range, then Gosu throws an exception.

For example, if you removed an auto from an auto policy before the slice date, Gosu throws an exception because that auto is not effective at that date. However, if you call `getSlice` on the `PolicyPeriod` and navigate down the object graph for that slice, you do not need to worry about unavailable effective dates for subobjects. This is why it is typically best to call `getSlice` on the `PolicyPeriod` and navigate down the object graph from there.

**Note:** For many use cases, it is best to slice the root `PolicyPeriod` with a specific slice date. In other words, call `getSlice` on the root `PolicyPeriod` entity instance and then navigate down the object graph from there.

### Automatic Changes for Slice Mode Edits

When editing objects in slice mode, certain changes happen automatically:

- If in slice mode and you change an existing subobject of the `PolicyPeriod`, the changed entity instance automatically splits into two entities. One entity instance represents the change before the slice date and one entity instance represents after the slice date. The split does not occur if the split date is already the effective date of the entity instance. If a split is needed, it happens immediately in the in-memory version of the entity instance. Even if a property is reverted after a split, the entity instance remains split.
- If in slice mode and you add a subobject of the `PolicyPeriod`, the new entity instance automatically has an effective date of the slice date.
- If in slice mode and you delete a subobject of the `PolicyPeriod`, the entity instance automatically has an expiration date the day of the slice date. However, if the split date is already the effective date of the entity instance, the entity instance is simply deleted.
- To handle situations like out-of-sequence jobs and preempted jobs, merging changes forward to future effective dates is not handled automatically in Gosu at the time the change is made. The job process and workflow files detect this issue at quote time. The job files and PCF files offer the user a chance to apply changes to later effective dates and resolve any conflicts. For more information, see “Out-of-sequence Jobs” on page 501, “Preempted Jobs” on page 503, and “Applying Changes to Future Renewals” on page 507.

#### Slice Mode Notes

- You can mark an entity instance in the data model to not automatically split on an edit. For example, PolicyCenter uses this feature for objects that never split, such as transactions and forms. Other objects can be split but only split explicitly, for example costs and Workers’ Compensation Jurisdictions. To mark an entity instance to not automatically split on editing, set the data model attribute `autoSplit` to `false`.
- A scalable field that scales as a result of a split shows up in the PolicyCenter list of differences as a *window edit*.

## Window Mode API Overview

To view objects across all effective time, you can view the policy objects (or a whole branch) in window mode. Typically you do this with a version list for an object. A version list represents all versions of that one object within one policy period. For example, a version list for a car represents all versions of that car during the entire policy period.

**IMPORTANT** For important overview information about window mode and version lists, see “Slice Mode and Window Mode Overview” on page 489.

If you have a version list, you can get various information about the object. For example, get all versions of the object, or navigate up or down the hierarchy to other entity instances or version lists.

The most important APIs on a version list are as follows. All entities results are in window mode unless otherwise noted.

- The `versionList.AllVersions` property gets all versions of this object. All results are in window mode. The order in the list (the sort order) is the effective date for each version.
- The `versionList.asOf(date)` method gets the one version of this object on that date, or `null` if none were effective on that date.
- If an entity property contains an array of entity instances, Gosu generates two version list properties related to that original entity property:
  - Gosu generates a version list property whose name exactly matches the property name on the original object. It contains a list of all unique objects that are ever in that array at any effective time in the period. For example, a personal vehicle object contains its drivers in the `vehicle.Drivers` property. Thus, a vehicle’s version list also has a `Drivers` property. It contains a list that contains one version list for each unique driver for that vehicle. The items in this list have no defined order. Do not rely on the order.
  - Gosu generates a version list method whose name matches the property name on the original object but with the suffix `AsOf`. This method takes a date argument, which is an effective date. The `AsOf` method

returns a snapshot of that array property as of that effective date. The `AsOf` method converts and returns the results to a list, which is typically easier to code with than arrays. For example, a personal vehicle object contains its drivers in the `vehicle.Drivers` property. Thus, a vehicle version list has a `DriverAsOf(date)` method. This method returns a list of that vehicle's drivers on that date. The entities are returned in window mode. The items in this list have no defined order. Do not rely on the order.

**Note:** Generated methods like `DriverAsOf` are a rare exemption to the normal Gosu coding rule that method names always begin with a lowercase character. PolicyCenter capitalizes the first character in this case to improve Gosu code readability because these methods mirror the original property names with an initial capital letter.

Also, it is important to note that you can get a version list from any object by getting its `VersionList` property.

The following subtopics describe real-world tasks with a version list by using as an example a car (a `PersonalVehicle`) and its version list. For code examples, assume that the variable `vehicleVL` contains a version list for the car (`vehicle.VersionList`). There are three versions of this car in this period, each with a different color. The car's version list represents exactly three versions of this car.

### Get all versions of a particular car

To get all versions of this particular car in the policy period, use the following Gosu code:

```
var val = vehicleVL.AllVersions
```

If the car changed twice during the period, such as a color change, the result is a list with three entity instances. Each represents a different version of this car.

### Get the car version that is effective at specified date in window mode

To determine which version of this car (if any) was effective at a specific effective date, and return it in window mode, use the following Gosu code:

```
var vehOnDate = vehicleVL.asOf(date)
```

The result is a single car object returned in window mode, or `null` if no version of the car is effective at that date.

### Get the car version that is effective at specified date in slice mode

To get this car's version that was effective at a specific effective date, and return it ready to make slice mode changes at that date, use the following Gosu code:

```
var vehSlicedOnDate = vehicleVL.asOf(date).getSlice(date)
```

The result is a single car object returned in slice mode, assuming a car is effective at that date. If no car is effective at that date, it throws a null pointer exception, since `asOf` returns `null`. Because the code calls the `getSlice` method of the result of `asOf`, that is a method invocation on a `null` value.

**Note:** For more information about null-safety of properties but not methods, see “Property Paths are Null Tolerant” on page 194 in the *Gosu Reference Guide*.

### Get the set of all drivers who were ever drivers of this car

To get the list of all drivers who were ever drivers of this car during this period, use the following Gosu code:

```
var drivers = vehicleVL.Drivers
```

The result is a list. The list contains one version list for each unique driver of this particular car. Each version list represents a single driver. From each version list, you can get all entity instance versions of that unique driver by getting the version list's `AllVersions` property.

### Get the set of all drivers who were ever drivers of this car at specified date

To get the list of all drivers who were drivers at a specific date, use the following Gosu code:

```
var theDate = new Date() // a date object, in this case "today"
var drivers = vehicleVL.DriversAsOf(theDate)
```

The result is a list containing one or more `VehicleDriver` objects in window mode. Think of this as a snapshot of the entity array on that effective date, with all other entities hidden. The type of the result is `java.util.List<VehicleDriver>`.

#### Get the drivers of this car at specified date and return their contact public IDs

To get the list of all drivers who were drivers at a specific date, then get the public IDs for their contacts, use the following Gosu code:

```
var arrayOfPID = vehicleVL.DriversAsOf(date)*.PolicyDriver*.PublicID
```

The `DriversAsOf` method returns a list of `VehicleDriver` objects. Each one of those objects links to the actual contact for that driver through its `vehicleDriver.PolicyDriver` property. That is the object that contains the driver name and drivers license number. The Gosu array expansion operator `*`. extracts data from each item in an array or list, then returns results in a single-dimension array. Similarly, if you pass it a list, it returns a list. For details, see “List and Array Expansion (`*`)” on page 255.

## Safely Accessing Foreign Keys with Slice Mode

It is important to understand that a window mode entity instance has no slice date. It represents the car over a range of dates. With a reference to a window-mode entity instance, you can access simple properties such as a `String` or a number.

Generally speaking, with a window mode entity instance, do not access links to other entities by using standard foreign key properties on the object. For example, for a vehicle entity instance in window mode, do not access its `PolicyLine` property. In window mode, this property is unclear with respect to the policy line as-of date.

If you access foreign key fields on a window-mode entity instance, Gosu returns the value of that property as of the last second of the window mode object’s effective time. This is sometimes what you want, but in typical code it is not.

For example, suppose you have a window mode version of a car in a variable called `vehicleUnsliced`. The `vehicleUnsliced.GarageLocation` property returns the garage location as of one second before the expiration date of the `vehicleUnsliced` object.

**Note:** That date might not be the last moment before the expiration date of the car on the policy. It is only the last moment for this particular window mode object. There may be a version of this object with a later expiration date.

For typical code, do not rely on this feature to navigate to related objects since the return result is not typically what you want. Instead, it is typically best to convert the window mode entity instance to a slice mode entity instance and then access its related objects at that slice date.

Any objects that you access from it are now automatically in slice mode because you accessed them from a slice mode object.

Compare the following two code examples.

The following code slices a window mode vehicle and then gets its policy line at that date:

```
unslicedVehicle.getSlice(date).PolicyLine
```

The following code gets the `PolicyLine` property from an unsliced (window mode) version of a vehicle, and then slices that result. This result is potentially different from the previous example. That is because this relies on the window feature discussed earlier in the topic. Although it looks similar, this code may return a different result from the first example. This code gets the policy line as of the last moment of this car’s effective date range. Then, the code slices that policy line at the desired date.

```
unslicedVehicle.PolicyLine.getSlice(date)
```

The important thing to notice is that the two lines of code may access different policy lines entirely. The first one accesses second example gets the `PolicyLine` property as of an explicit date. The second one accesses the policy line as of an implicit date (one second before the expiration of that window mode entity instance).

Secondly, when you use the `getSlice` method, the date must be within the effective date range of that individual version of the object. With that in mind, notice that in the first example, the date must be within the effective date range of the unsliced car object. In the second example, the date must be within the effective date range of the policy line.

It is important to keep track of which entity instance is most appropriate to call `getSlice` on. If you do not know whether the current version is the correct one, get the version list and call its `asOf` method:

```
unslicedVehicle.VersionList.asOf(date).getSlice(date).PolicyLine
```

---

**IMPORTANT** Generally speaking, on a window mode object be careful with directly accessing any foreign key references or array references. If you access a foreign key or array property on a window mode entity instance, Gosu returns the value as of one second before the expiration date of that object. In typical code, this is not what you want. Instead, get the version list, then get the correct window mode version of the object, and then slice it at an explicit date. Carefully review the Gosu code examples in this topic.

---

## Version List API Reference

The following table contains a list of methods on the version list where there is exactly one of each, independent of what properties are on the entity. The following table assumes an example entity instance of type `PersonalVehicle` with a property called `Drivers` of type `PolicyDriver`.

The version list is specially typed to match the entity instance for which it is a version list but with the suffix `VersionList`. For example, a version list for a `PersonalAutoLine` object has type `PersonalAutoLineVersionList`.

Every version list has properties and methods that are common to all version lists. Additionally, from the Gosu type system it automatically has properties that mirror property names on its source object. For a detailed explanation of these mirrored property names, see “Version List API Reference” on page 494.

To do this task	Use this property or method	Returns	Return type for Our Example
Get all versions of this entity instance in this period.	<p>AllVersions</p> <p>This property gets all versions of this entity instance across effective time in this contractual period. Each entity instance reference is set to edit in window mode. This is a very important version list property in real-world PolicyCenter code. For example, use this in rating code to iterate across all effective time versions of an object that you need to send to the rating engine.</p> <p>This is an important method if you want to access properties on an entity instance and the property is not an array property. You can iterate across all versions and get the desired property from each one.</p> <p>Note: When getting this property on a version list for a <code>PolicyPeriod</code> (the graph root), this property contains only one version since this root entity instance is not revised.</p>	<p>A list of objects of the original entity type. Each entity instance has different effective date ranges (and they do not overlap).</p> <p>Each entity instance is in window mode.</p> <p><b>IMPORTANT:</b> This is a very important version list property for real-world code.</p>	<code>List&lt;PersonalVehicle&gt;</code>
Get the version of this object that is effective at a particular date, if any such version exists.	<p>asOf(date)</p> <p>This method returns this entity instance as of a particular date passed as a method argument.</p> <p>This is an important method if you want to access properties on an entity instance where the property is not an array. This is because you can choose a date to pass to this method, and get the desired property from the result.</p> <p>This method is similar to the property in the first row in the following table, whose name in our example is the <code>Drivers</code> property. However, this uses <i>type system reflection</i> to get the property (pass the property name passed as a property argument) and a slightly different type return value.</p>	<p>One instance of the original object type, returned in window mode.</p> <p><b>IMPORTANT:</b> This returns null if no version of this object is effective at that date. For example, the object might be removed, canceled, or not yet added to the policy.</p>	<code>PersonalVehicle</code>
Check if an object has effective-date gaps.	<p>HasGaps()</p> <p>This property checks means that the entity instance has at least some amount of non-effective time between two ranges of effective time in that contractual period.</p>	<code>boolean</code>	<code>boolean</code>

To do this task	Use this property or method	Returns	Return type for Our Example
Check if an object has overlapping duplicates across effective time.	<code>hasOverlaps()</code>  If true, some code created invalid data. This is most likely due to incorrect manipulation of entities in window mode. PolicyCenter has built-in validation routines that use this method to detect certain types of problems to fix before binding the branch. You can choose to use this method in your own validation code or other Gosu code.	<code>boolean</code>	<code>boolean</code>
Get all versions of this object, but typed to the root of all revised entities.	<code>getAllVersionsUntyped</code>  This is similar to the <code>AllVersions</code> property, but with a different return type declaration and it is a method instead of a property.	A list of effective dated entities. The type for each item is the root class of all revised entities, which is <code>EffDatedBean</code> . In typical code, you probably need to cast each item to the specific entity sub-type.	<code>List&lt;EffDatedBean&gt;</code>
Gets the entity instance as of a date, but typed to the root of all revised entities, <code>EffDatedBean</code> .	<code>getVersionAsOf(date)</code>  This is basically the same as the <code>asOf</code> method, but with a slightly different return type.	One entity instance typed as the root of all revised entities ( <code>EffDatedBean</code> )  If you pass an invalid date for this method, Gosu throws an exception.	<code>EffDatedBean</code>

The following table contains a list of methods on the version list. There is one of each of these members for every property on a policy entity that contains arrays of other entities. This commonly represents navigating down a hierarchy, but that is not always the case. For example, a policy object contains an array of policy lines, a vehicle object contains an array of drivers. This applies only to directly database-backed properties, not to enhancement properties or to any methods.

To make it easier to understand the method and property names with concrete example, the following table uses example entity PersonalVehicle and its property called Drivers. Each driver has the type PolicyDriver.

To do this (with a property that contains an array of entities)	Use this property or method	If vehicle property Drivers, the version list property is called	Returns	The return type for our PersonalVehicle example
Get all version lists for this property.	The same name as the property in the source object.	Drivers There is a method like this for each property on the original object that contains an array.	A list of version lists, typed to the property type on the source object with a suffix of VersionList.  Each version list in the result represents a unique entity instance (a shared fixed ID) and all its versions for this period.	List<PolicyDriverVersionList>
Get contents of this property as of a particular date.	Add the AsOf suffix to the name of the property in the source object. This is a method (even though its first character is capitalized).  Pass the date as an argument.	DriversAsOf(date) There is a method like this for each property on the original object that contains an array.	List of the type contained in the array property. The return value is a list, not an array.  This could be an empty list if no child objects in that array are effective at that date due to being removed or canceled.  This might return an empty array, but not null and not throwing an exception for this condition.	List<PolicyDriver>
Add an object to the version list that represents a property that is array of objects.	The property is called addTo and then the source property name.  This method takes an entity instance of the type of the original property (in our example, <b>PolicyDriver</b> ).  This change technically always happens in window mode. Because of this, effective and expiration dates on the entity instance are preserved.  In contrast, when adding entities in slice mode, PolicyCenter overrides the effective date with the slice date.	addToDrivers(obj) There is a method like this for each property on the original object that contains an array.	No return result.	void

To do this (with a property that contains an array of entities)	Use this property or method	If vehicle property Drivers, the version list property is called	Returns	The return type for our PersonalVehicle example
<b>The following methods use type reflection (dynamic access) and are rare in typical code. Use these methods only for algorithms that are impossible with typesafe APIs, such as if you do not know property names at compile time.</b>				
Get all version lists for this property, but use reflection to specify the property name.  <b>IMPORTANT:</b> Reflection APIs prevent Gosu from checking the correctness of arguments at compile time.	getArray(propName)  This method is similar to the property mentioned in this table with the row with the label "Get all version lists for this property." on page 497 and example name Drivers.  However, this uses <i>type system reflection</i> to get the property. Pass the property as an argument. For the argument, see the example "Get all drivers of the car, but use reflection APIs" on page 499 for syntax.	getArray(propName)  For the argument, see the example "Get all drivers of the car, but use reflection APIs" on page 499.	A list of version lists. Each version list in the result represents a unique entity instance (a shared fixed ID) and all its versions for this period.  The result type is List<EffDatedBean>, not a more specific subtype. At compile time, Gosu does not know the type of the results. Your code must cast each list member to your desired subtype.	List<EffDatedVersionList>
Get contents of this property as of a particular date, but use reflection to specify the property name.  <b>IMPORTANT:</b> Reflection APIs prevent Gosu from checking the correctness of arguments at compile time..	getArrayAsOf(propName, date )  This method gets a list of the objects that are effective in the area at the particular date.  This method is similar to the property mentioned in this table with the row with the label "Get all version lists for this property." on page 497 and example name Drivers.  However, this uses <i>type system reflection</i> to get the property. Pass the property as an argument. For the argument, see the example "Get all drivers of the car, but use reflection APIs" on page 499 for syntax.	getArrayAsOf(propName, date )  This method is similar to the property mentioned in this table with the row with the label "Get all version lists for this property." on page 497 and example name Drivers.  However, this uses <i>type system reflection</i> to get the property. Pass the property as an argument. For the argument, see the example "Get all drivers of the car, but use reflection APIs" on page 499 for syntax.	A list of the objects that are effective at the particular date. The result type at compile time is List<EffDatedBean>, not a more specific subtype. At compile time, Gosu does not know the type of the results. Your code must cast each list member to your desired subtype.  This could be an empty list if no child objects in that array are effective at that date due to being removed or canceled.  If you pass an invalid date for this method, it returns an empty array. This does not return null and does not throw an exception for this condition.	List<EffDatedBean>  Typical code needs to cast each version list to a more specific subclass such as List<VehicleDriverVersionList>.

## Advanced Version List Examples

The following examples use a variety of PolicyCenter objects and includes some more advanced `VersionList` APIs. For version list APIs that return lists or lists of version lists, you can use the powerful Gosu enhancements for collections to make Gosu code as concise as possible. For example, you can use code such as

```
vehicle.VersionList.AllVersions.first()
```

Many collection enhancements have arguments that are Gosu blocks, which are in-line functions that make powerful Gosu code easy to read.

For more information about collection enhancements, see “Collections” on page 251 in the *Gosu Reference Guide*. For more information about blocks, see “Gosu Blocks” on page 231 in the *Gosu Reference Guide*.

### Get all drivers of the car, but use reflection APIs

To get the version lists for all unique drivers of a vehicle, but specify the property name by using reflection (dynamic access at run time):

```
var driversProp = PersonalVehicle.Type.TypeInfo.getProperty("Drivers") as gw.entity.IArrayPropertyInfo
var driversArray = vehicleVL.getArray(driversProp)
```

### Get all costs from an auto policy line

For example, suppose you have a `PersonalAutoLine`. If you want to get all its personal auto costs (in its `PACosts` property) in the contractual period, use the Gosu code:

```
myCostVersionLists = autoPolicyLine.VersionList.PACosts
```

This returns a list of version lists. Each of these version lists represent **one cost** and all its costs across effective time.

It is an extremely common mistake to use code that looks like

```
notAllCosts = autoPolicyLine.PACosts // generally NOT what you want to do
```

This code does not get all the costs. It gets only the costs associated only at the slice date (which typically is meaningless). You usually want **all** the costs for the policy period, which represents the total price of the policy.

### Extract all costs from an auto policy line and return them in a 1-dimensional array

To extract all cost entities across effective time, use the `flatMap` collection enhancement method. As an argument it takes a Gosu block. In this case, the block takes a cost version list as an argument. Then the code gets all versions of this cost. Then finally the `flatMap` enhancement method combines them into a single list.

```
var allCosts = autoPolicyLine.VersionList.PACosts.flatMap(\ costVL -> costVL.AllVersions)
```

The result is a list that contains all auto costs as one flattened list.

Some cost objects have the same fixed IDs as other costs (they are the same cost) but vary in effective dates.

### Original drivers of a vehicle

Get the drivers of the vehicle as of the earliest effective date on the policy for this car:

```
var firstVersionUnsliced = vehicleVL.AllVersions.First()
var origDrivers = firstVersionUnsliced.getSlice(firstVersionUnsliced.EffectiveDate).Drivers
```

The list that `vehicle.AllVersions` returns is ordered by effective date. Getting the first item from the list (as this example does) gets the item with the earliest effective date.

### Get all coverages on a vehicle and print data from each version, segregated by each unique coverage

Display all the coverages that any time were on the vehicle along with display name and the date range covered by that version:

```
for (covVL in vehicleVL.Coverages) {
 for (cov in covVL.AllVersions) {
 print("${cov.Pattern.DisplayName}: effdate ${cov.EffectiveDate}, expdate ${cov.ExpirationDate}")
 }
}
```

### Get a car's garage location at a specific date

Suppose you want to get a car's garage location. Since the location might have changed, you often just want to show one location, usually the last one or perhaps the one as of a particular date. This example assumes you have a vehicle version list in a variable called `vehicleVL`.

```
// find out *which* version of this object was effective on that date
var vehicleUnsliced = vehicleVL.AsOf(asOfDate)
if (vehicleUnsliced == null) throw "No vehicle effective on that date"

// get the version of the garage at that date
// Get the Garage in slice mode, just before the end of its active time
Var vehicleSliced = vehicleVersion.getSlice(vehicleUnsliced.ExpirationDate - 1 sec)
var g = vehicleSliced.garageLocation

// print the location (in real world code, display in PCF files instead)
print("garage ${g.AddressLine1} / ${g.AddressLine2} / ${g.City} / ${g.State} ")
```

See “Safely Accessing Foreign Keys with Slice Mode” on page 493 for related discussion.

### Get all available drivers for a car at a specific date

This example assumes you have a vehicle version list in a variable called `vehicleVL`.

```
var vv = vehicleVL.AsOf(asOfDate).AvailableDrivers
```

Although `PersonalVehicle.AvailableDrivers` returns an array of driver objects, it is an enhancement property not a database-backed property. The version list properties that Gosu creates for array properties (such as `vehicle.DriversAsOf(date)`) only exist for database-backed properties. Thus, it might seem like the following Gosu code works, but it is a compile error:

```
var v2 = vehicleVL.AvailableDriversAsOf(asOfDate) // compile error!
```

## Working with Window Mode (Unsliced) Objects

If you have reference to any revised subobject of `PolicyPeriod`, simply get its `Unsliced` property to get a reference to that entity instance in unsliced mode. Remember to `save` the return result. The original reference is unchanged.

**Note:** The `PolicyPeriod` entity instance is not a revised entity instance. It is the root of the graph of revised objects and it does not have an `Unsliced` property.

### Version Lists Always Return Entities in Window Mode ('Unsliced' Mode)

If you get a version of an entity instance from a version list, Gosu always returns that entity instance as a window mode entity instance. In other words, you have the entity reference but no implicit slice date. This means that the entity instance represents the car over a range of dates.

With an unsliced entity instance, you can access simple properties on the object but you cannot access links to other entities (foreign keys and arrays of entity instances).

### Window Mode Entities from Database Queries

The typical way to get window mode versions of entities is to use version lists. However, there are other ways to get a window-mode reference to an entity instance if you have a slice mode reference to an entity instance.

The most common other way to reference an entity instance in window mode is an entity instance that you retrieve through a database query. See “Query Builder APIs” on page 125 in the *Gosu Reference Guide* for related information.

Database query results do not have any inherent context from which to determine a slice date. Thus, by default any objects from a database query return in window mode (and in a read-only bundle). It is usually correct to then convert the item into slice mode as of a certain date with the version list `getSlice` method:

```
var sliceModeVehicle = myVehicle.getSlice(sliceDate)
```

However, it depends on the context. In some cases, you might want to work with the object in window mode.

### Naming Conventions

When reading code, you may get confused as to whether you are working with sliced or unsliced objects. One approach to improving code readability (and reducing coding errors) is to consistently name variables. For variables that contain unsliced objects, include the suffix `Unsliced`. For example, `policyLineUnsliced`.

The convention is that variables that do not have the `Unsliced` suffix contain a sliced version (the more common case).

For example:

```
var firstVersionUnsliced = vehicleVL.AllVersions.First()
var drivers = firstVersion.getSlice(firstVersion.EffectiveDate).Drivers
```

## Comparing Window Mode Edits to Slice Edits

If you directly edit an entity instance in window mode, the changes affect the full effective period **only** of that one entity instance.

In contrast, a slice mode edit using the default split-on-edit approach, in which changes affect the entity instance from the edit date forward. Unlike slice editing, window mode changes do not merge changes forward when the object is edited in window mode. For example, when an out-of-sequence policy change is merged forward, the window mode edits do not merge forward in effective time. For more information about out-of-sequence changes, see “Out-of-sequence Jobs” on page 501.

For example, Workers’ Compensation line of business includes Workers’ Compensation payroll amounts that do not change across effective time. A change to these amounts applies to the entire contractual period. People need to provide the amount of payroll that they had for a class of workers within a couple of separate date ranges. They need to edit these numbers directly, not partway through the period. They need to see all date ranges at once, not just the ones effective as of a given date. It would make no sense to merge a change made to the first date range forward to a later one. You must edit each one separately.

Also note that editing in window mode disables automatic scaling and splitting behaviors that would normally happen in slice mode.

Other objects that PolicyCenter edits in window mode include `RatingPeriodStartDates` and `WCCoveredEmployees`.

## Out-of-sequence Jobs

Many policy jobs have effective dates later than the effective date of any existing bound revisions for that contractual period. The change implicitly applies from the job’s effective date until the end of the contractual policy period. For example, increasing coverage on an effective date applies for the rest of the contractual period, or canceling a policy is effective for the rest of the policy period.

However, if a change is bound to take effect before a previous change (that is, earlier in effective time), there are additional implications for completing this change. Depending on what changes already happened to the policy, sometimes PolicyCenter requests that you review how to apply changes for the rest of the contractual policy period.

For example, suppose the following standard order of changes:

1. On January 1, the customer adds new auto policy effective all year for a red car, covered for \$10,000. The effective date of this change is January 1.
2. On March 1, customer increases a specific coverage on the car to \$20,000, effective from that day to year end. The effective date of this change is February 1.
3. On March 2, the customer calls to say that the original car was painted blue on February 1. The effective date of this change is March 1. This effective date is later than the effective date of the previous change.

This is a regular change because effective dates of the changes are later than effective dates of previous changes.

However, if you reverse the last two effective dates, the order of changes would be:

1. On January 1, the customer adds new auto policy effective all year for a red car, covered for \$10,000. The effective date of the change is January 1.
2. On March 1, the customer increases a specific coverage on the car to \$20,000, effective from that day to year end. The effective date of the change is March 1.
3. On March 2, the customer calls to say that the original car was painted blue on February 1. The effective date of the change is February 1. This effective date is earlier than the effective date of the previous change.

The last change in that example is an out-of-sequence change because February 1 is earlier than March 1. For effective time after February 1, there are two date ranges:

- From February 1 to February 28 in effective time, the **PolicyPeriod** must represent the newly painted blue car with the original coverage.
- From March 1 to year end in effective time, the **PolicyPeriod** must represent the updated increased coverage. However, PolicyCenter considered this a red car for this time range before the latest change. Was the car blue for the rest of the year or did it change only from February 1 to March 1?

Any change with effective date ordering like this is called an *out-of-sequence job*. Any PolicyCenter job, such as cancellation and reinstatement, not just policy change jobs can be out of sequence. A job is out of sequence if its effective date is earlier than other jobs bound on the policy for that contractual period.

PolicyCenter automatically detects out of sequence jobs. Some changes may not need user intervention. In other cases, you must review out-of-sequence conflicts in the policy review tab called **Out-of-Sequence Conflicts** before binding the job.

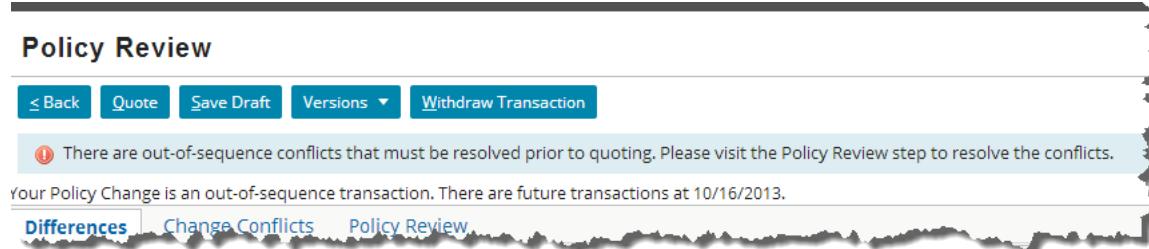
## Out-of-sequence Job User Interface

When PolicyCenter detects a job as out of sequence, PolicyCenter warns you when you start the job.

If you continue with the out-of-sequence transaction, there might be changes you need to review to determine whether to merge that change forward for the rest of the contractual period.

If there are no out-of-sequence conflicts, the **Policy Review** page for the job contains only the same differences it would show if it were not out of sequence.

However, if there are out-of-sequence issues, PolicyCenter alerts you to this issue in a yellow box at the top of the window when you try to bind the job:



The screenshot shows the Policy Review page. At the top, there is a navigation bar with buttons for Back, Quote, Save Draft, Versions, and Withdraw Transaction. Below the navigation bar, a yellow warning box displays the message: "There are out-of-sequence conflicts that must be resolved prior to quoting. Please visit the Policy Review step to resolve the conflicts." Underneath the warning box, a status message says: "Your Policy Change is an out-of-sequence transaction. There are future transactions at 10/16/2013." At the bottom of the page, there is a tab navigation bar with three tabs: Differences, Change Conflicts, and Policy Review. The Change Conflicts tab is currently selected.

From there, first click on the **Policy Review** tab.

Next, click on the **Change Conflicts** tab to review each change that conflicts with another job with later effective date. You can choose to override (merge) later-effective-date jobs with your recent change. Overriding later-effective-date jobs has the effect of merging forward your change for the rest of the contractual policy period.

You can choose to override all conflicts or none by using the **Override All** or **Override None** buttons. Finally, click the **Submit** button to finish the process.

Or, select the radio button Yes or No for each conflict. Finally, click the **Submit** button to finish the process.

The screenshot shows the 'Policy Review' screen. At the top, there are navigation buttons: 'Back', 'Quote', 'Save Draft', 'Versions', and 'Withdraw Transaction'. Below them is a message: 'Your Policy Change is an out-of-sequence transaction. There are future transactions at 10/16/2013.' A tab bar includes 'Differences', 'Change Conflicts' (which is selected), and 'Policy Review'. A warning message states: 'Some values specified in this Policy Change conflict with changes made with future effective dates. Please indicate which future conflicts are to be overridden, and then click Submit.' A note below says: 'Warning: Submitted changes cannot be undone.' Two buttons are highlighted with red boxes: 'Override All' and 'Override None'. A table lists a conflict: 'Uninsured Motorist - B...' with an effective date of '15/30' and a count of '250/500'. To the right of the table is a checkbox group for 'Override Future Conflicts' with options 'Yes' and 'No', also highlighted with a red box. A 'Submit' button is at the bottom left.

After you submit your change conflict choices by using the **Submit** button, simply requote and bind the job as usual.

## Back-dated Versus Out-of-sequence Job

Not all out of sequence changes are back-dated, and not all back-dated changes are out-of-sequence. These are really two separate issues, which in some cases may both be true, but neither one necessitates the other.

Compare these two definitions:

- A job is back-dated if the change has an effective date earlier than today.
- A job is out-of-sequence if the change's effective date is earlier than the effective date of another job in that contractual period.

## Validation Issues and Out-of-sequence Jobs

If there are validation issues with a job, PolicyCenter requires you to handle validation issues before it displays the out-of-sequence user interface. In most cases, fixing the validation issues at the effective date of this job is sufficient to prevent validation issues when applying (merging forward) changes in the **Out-of-Sequence Conflicts** tab.

In very rare cases, even after handling out-of-sequence conflicts there can be validation errors in the same contractual period with later effective dates. For example, suppose you changed driver usage time percentages and caused out-of-sequence conflicts and applied some driver usage values forward but not others. This would mean you could temporarily cause the driver percentages to be less than 100% or greater than 100%. If validation errors at future effective dates occur due to handling out-of-sequence conflicts, PolicyCenter displays a special user interface to handle this case.

PolicyCenter alerts you to the validation errors and displays a picker in the left navigation bar so you can select future effective dates at which there are validation errors. After selecting a future effective date that contains the validation error, you can update the values that cause the validation errors.

## Preempted Jobs

Although some PolicyCenter jobs start and finish quickly, other job take a long time to complete the entire life cycle. Sometimes this delay is due to technical reasons such as contacting external systems. Sometimes this delay is required for business reasons such as renewal jobs that might start months before the renewal effective date. There may also be legal reasons such as legally-enforced delays during cancellation.

When jobs take a long time to complete, chances increase that multiple jobs started on the same branch and are in process at the same time. For instance, two jobs are based on exactly the same **PolicyPeriod** entity instance and its subobjects. When the first job finishes there is no problem. When later jobs started at the same time complete,

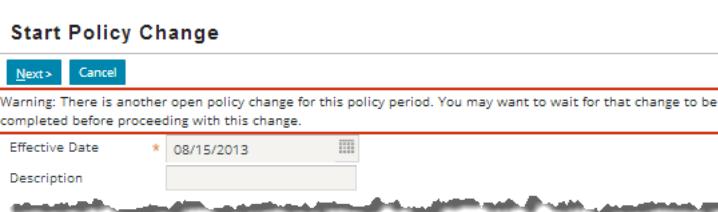
there may be challenges binding the new changes. The job that finishes second does **not** have the changes recently made and bound by the job that finished first.

When two jobs run concurrently like this, this situation is called *preemption* when the second job to finish attempts to bind. When PolicyCenter tries to bind a preempted branch, initially the branch does not contain preempted changes. PolicyCenter must incorporate the changes from the preempting branch. Preemption applies to any PolicyCenter job, such as cancellation and reinstatement, not just policy change jobs. After the first job to finish is bound, any unfinished jobs are preempted. PolicyCenter attempts to fix these problems early, as soon as you view a preempted job rather than just waiting until the preempted job tries to bind.

For example, suppose on a personal auto policy, two users start policy change jobs at the same time:

- One policy change adds an additional vehicle, effective March 1, keeping coverage amounts the same
- One policy change increases the coverage amount on the original car, effective April 1. Remember that this policy was based on the original legally-enforced policy when the policy change started. PolicyCenter represents this policy change as a branch (a *PolicyPeriod* entity instance and its subobjects) that is a clone of the original branch before any of the recent changes.

PolicyCenter detects potential preemption when starting a job if it appears that another job is in progress. The second concurrent job to start displays a warning in the user interface:



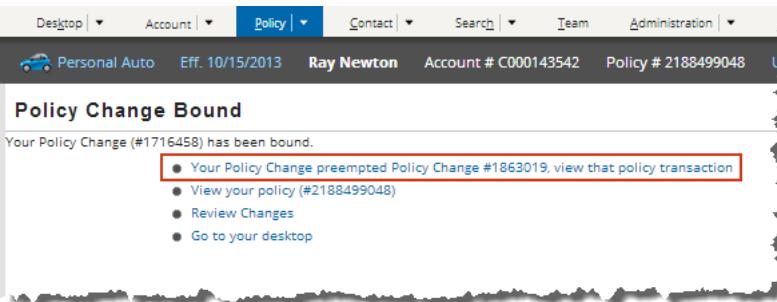
This warning does not indicate that a preemption will necessarily occur, or that it already occurred. However, if both branches eventually bind, one of the two jobs will be preempted.

Despite this warning, PolicyCenter lets the user start the policy change job or another job anyway. The complexity of preemption really takes place when the jobs finish or you try to handle the preempted job. The first change to finish preempts (takes precedence over) any concurrent changes not yet finalized.

**Note:** The complexity of preemption occurs after the first branch is bound and the user tries to work with the preempted non-bound job. This can happen in any phase of the preempted job, not just in the bind phase. The finish time determines which branch needs special handling, not the start time of the two jobs.

For example, if the policy change that adds the additional vehicle finalizes first, PolicyCenter makes that branch the legally enforced branch for this period. Nothing very unusual happens from a database or user interface perspective as part of binding this job.

However, after the user binds this change, if there are open jobs on this policy, PolicyCenter displays a warning this job preempted another transaction. It offers a link to view that policy transaction immediately.



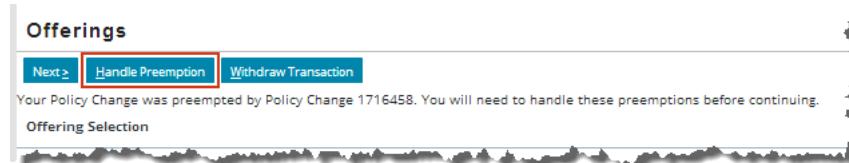
At this point in the application, it is possible that the user might withdraw or ignore other non-bound branches.

However, if any user attempts to bind the second change (the coverage increase change), that change was preempted. That change was originally based on a branch that is no longer the most recently bound branch. PolicyCenter displays special options to handle the preemption.

Let us first consider the case in which the job that you bound first in real-world time had an earlier effective date than the second-to-bind change. This is a standard preemption. Before binding the coverage increase change, PolicyCenter must add the additional vehicle to the draft branch containing the coverage increase before attempting to bind the increased coverage.

**Note:** PolicyCenter must merge changes like this during preemption. Otherwise, when you bind the coverage change, the additional vehicle would be missing. It would appear as if you removed the vehicle from the policy as of April 1 as part of the recent change even though that was not your intention.

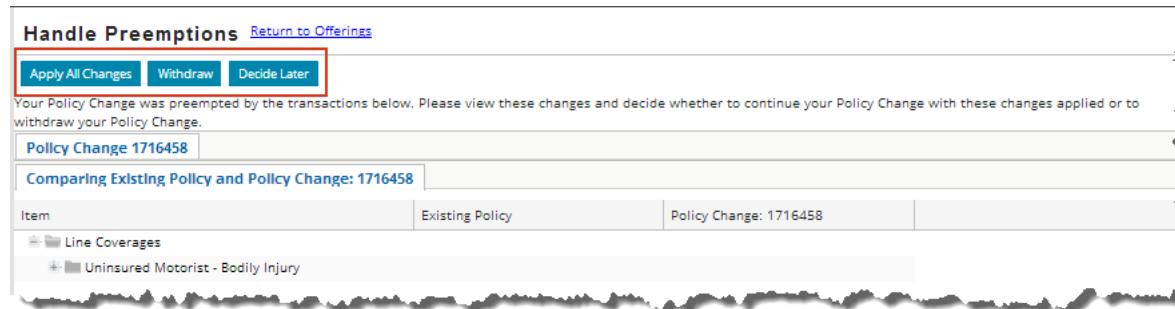
If you later view the preempted job, PolicyCenter warns you with a message at the top of the window. Also, the **Handle Preemption** button appears if you have preemptions to handle on this job.



You must choose to do one of the following:

- Click **Handle Preemption** to merge changes as appropriate from recently-bound jobs into the active job that is about to be bound.
- Click **Withdraw** to withdraw (abort) the current job.

If you click the **Handle Preemption** button to handle the preemption, you see the following screen.



You can use the tree navigation (clicking on - and + signs) to hide or show parts of items in the hierarchy.

Use the buttons at the bottom of the screen to choose among the following actions:

- Click the **Apply All Changes** button to apply all changes.
- Click the **Withdraw** button to withdraw the job.
- Click the **Decide Later** button to save the draft and return to the policy review screen.

If you apply changes and the **effective date** of your draft branch is **later** than effective dates of other bound jobs for that period, PolicyCenter simply applies the changes. This is a standard preemption.

However, if the **effective date** of your draft branch is **earlier** than effective dates of any bound jobs for that period, this preempted job is also out of sequence. Because it is out of sequence, there might be out-of-sequence merge conflicts. If there are merge conflicts, they appear in the **Change Conflicts** tab. You can review each conflict and determine whether to merge each change forward. For more information about the meaning of out-of-

sequence in PolicyCenter, see “Out-of-sequence Jobs” on page 501.

---

**IMPORTANT** A job can be both preempted and out-of-sequence, depending on the effective date of the current branch compared to the effective date of other bound branches. If you apply changes to handle the preemption, there might be merge conflicts. If this happens, PolicyCenter displays the same change conflicts user interface (the Change Conflicts tab) as a standard out-of-sequence job.

---

### How PolicyCenter Actually Handles Preemptions

In the PolicyCenter interface for preemption handling, it appears as if PolicyCenter simply merges changes into the latest selected (draft) branch. However, from a database perspective, PolicyCenter actually creates a new branch to handle the preemption.

The changes happen in the following order:

1. PolicyCenter creates a new branch that is a copy of the most recently bound **PolicyPeriod** in that contractual period. By definition, this includes all changes from any preempting branch (or branches for multiple preemptions). This is the safest way to preserve consistency with a legally enforced branch.
2. PolicyCenter then merges the changes you attempted to make in the preempted branch to this new branch.

---

**IMPORTANT** In rare cases, PolicyCenter cannot automatically reapply the changes. For example, if you make a change to a vehicle removed in a preempting branch. There is no longer a vehicle PolicyCenter can modify in the new merged branch. If such rare cases occur, after PolicyCenter re-applies changes, PolicyCenter opens a worksheet to notify you about change conflicts. This is just a notification and there is no required action to perform afterward.

---

3. PolicyCenter discards the branch that the user was actively working on (the preempted branch) after handling the preemption. PolicyCenter replaces it with the new merged branch in the user interface and in the database.
4. You can customize application logic that occurs after handling a preemption but before discarding the draft branch and binding the new **PolicyPeriod**. For more on this topic, see “Customizing Behavior After Handling a Preemption” on page 232 in the *Integration Guide*.

## Preemption Reduction Configuration

To reduce preemption and to allow customer flexibility, PolicyCenter provides configuration parameters in your main `config.xml` file:

- `DelayRenewalIfConcurrentPolicyChangeInProgress` - Specifies whether a Renewal job must wait when a policy change job is in progress for the expiring policy period. This delays starting the renewal so that you create the copy after the change is completed.
- `MaxRenewalDelayDueToConcurrentPolicyChange` - Maximal amount of time to delay a renewal due to a concurrent policy change job. If empty (the default), renewal is delayed until all concurrent policy change jobs complete. Specified as a time duration specification parsable by `DateTimeUtil.parseTimeDurationSpec()`. For example, "5d" represents a delay of 5 days. This is sometimes considered a preemption configuration property, however this is not real preemption. For related issues, see the section “Applying Changes to Future Renewals” on page 507.

There is no generalized version of locking a policy or a branch. Because cancellations and renewals sometimes take a long time, strictly locking entire policies or policy revisions would disrupt other jobs if they were in process. Instead, PolicyCenter warns when jobs are started in the user interface, and lets users handle preemption conflicts (if they exist) when jobs are finally bound.

## Applying Changes to Future Renewals

If a user changes a policy in the current period but a future renewal revision already exists, PolicyCenter displays a special screen. That screen asks the user whether to apply (merge) changes forward to the renewal period.

If you do not apply the changes from the current job to the renewal period, you can apply changes later when you revisit the policy change job. PolicyCenter asks again whether to apply these changes to the renewal period.

If the user says yes, PolicyCenter reapplies the changes to the future renewal branch. However, the behavior is slightly different depending on whether the future renewal contractual period is bound or unbound:

- If the future renewal period is bound, a new job is started in the renewal period to apply the changes. The new job in the renewal period always is a policy change job based on the most recently bound branch in the renewal period. The changes only include the ones you made in the prior period. If multiple jobs in the prior period are bound and the user requests to apply these changes to the renewal period, multiple jobs are created in the renewal branch. The new jobs are independent of each other (some might be bound, some might be withdrawn, as desired).
- If the future renewal period is unbound, changes are made directly to the draft renewal branch. No new job is created to handle this case. If multiple jobs in the current (pre-renewal) period are bound and you request to apply changes to the renewal period, all changes are applied to that same draft renewal branch.

**IMPORTANT** In rare cases, PolicyCenter cannot automatically reapply the changes. For example, if you make a change to a vehicle removed in a preempting branch. There is no longer a vehicle PolicyCenter can modify in the new merged branch. If such rare cases occur, after PolicyCenter reapplies changes, PolicyCenter opens a worksheet to notify you about change conflicts. This is just a notification requires no action.

## Revisioning Rewrite Jobs

A rewrite job starts from a copy of the canceled policy. In most cases, the rewrite gets a copy of the policy at the cancellation date. When there is a lapse in coverage, the copy comes from the canceled portion.

There are several special different conditions to keep in mind:

- If a rewrite starts after the cancellation date, it gets a duplicate of the canceled policy as of the start date of the rewrite. In this case, the copy of the policy as of the cancellation date can have multiple slices, such as if the rewrite is out of sequence to other policy changes.
- If the start date is after the original end date, PolicyCenter duplicates the rewritten policy from the last day of the canceled policy. In this case, there is a single slice. Everything on the rewrite job has the same effective and expiration dates. This is, however, a rare case.

If the start date of a `PolicyPeriod` moves forward to a later date, PolicyCenter moves the effective date of all objects on the policy graphs forward to that date. All information about the original start date of the `PolicyPeriod` and its subobjects start date no longer appears as data in the `PolicyPeriod` graph. This is the intended and defined behavior. However, in some edge cases the result can be difficult to understand and can look strange or incorrect, so keep in mind how it works.

For example, suppose the following sequence occurs:

- You create personal auto policy, one vehicle, one driver, 9/1/09 through 3/1/10.
- You change the policy, effective 12/1/09, adding a second vehicle.
- You start a midterm rewrite, with the effective date initially set to the cancellation date (the default), 11/5/09.
- You bind this policy as is then in the policy term that resulted. Vehicle 1 has an effective date of 11/5, and vehicle 2 has an effective date of 12/1 (correctly). However, you might expect a one month lapse in coverage, so you change the effective date of the rewrite to 12/5/09, and save the draft.

- If you bind the policy at this point, on the policy term that results, both vehicle 1 and vehicle 2 have an effective date of 12/5 (correctly).
- However, you decide there was not supposed to be a lapse in coverage, so you change the effective date of the rewrite back to 11/5. This is unusual but possible. That branch in PolicyCenter no longer has the information about what the PolicyPeriod looked like before 12/5. Thus, PolicyCenter stretches back the PolicyPeriod to make everything that has an effective date of 12/5 have an effective date of 11/5. This results in vehicle 1 and vehicle 2 having an effective date of 11/5, which might seem incorrect but is the defined behavior in this case.

## Summary of Revisioning Terminology

The following table summarizes important revisioning terminology.

Term	Description
revisioning	How PolicyCenter tracks changes to a graph of objects in a policy through time, through both model time and effective time
branch (a policy revision)	The graph of objects with a PolicyPeriod entity instance at the root. Collectively a branch represents the truth of all effective dates in a contractual period as of the time it was made legally binding.
contractual period	A single policy term from the date the policy goes into effect (the effective date) to the date it expires (the expiration date). Generally speaking, a policy cannot have contractual periods that overlap in effective time, although if a policy is canceled or rewritten, contractual periods in a policy could overlap.
bound (promoted)	A branch that was made legally enforced, also known as legally binding.
model time	The real-world date and time that a version of the policy (or other object) was bound.
effective time	When something is relevant and enforced within a contractual period, independent of the model time. For example, if a year-long auto policy is canceled as of August 1, the effective date for the auto policy is January 1 through July 31. This is true independent of the date this change happens in model time.
branch ID and branch value	Foreign key to the PolicyPeriod entity instance that contains this entity instance. Within the same branch, all entities must share the same branch value. This value must be non-null. Gosu exposes this value as the BranchValue property, although the database column name is BranchID. If you use the query builder APIs, specify this property as BranchValue, not BranchID. For more information, see the BranchValue row in the table in “Revisioning Properties on PolicyPeriod Subobjects” on page 511
fixed ID	This ID describes one revised entity instance in multiple branches, or more than once in a branch with different effective/expiration dates. For example, suppose you need to change a car license plate number. The database contains two rows for the car: one for before the change, one for after. Both rows have the same fixed ID so that the system knows that it is two versions of the same car, not two different cars.
slice mode	Viewing a PolicyPeriod entity instance’s subobjects at a specific effective date, hiding entities that are not effective at that date. See “Slice Mode and Window Mode Overview” on page 489.
window mode	Viewing a PolicyPeriod entity instance’s subobjects, accessing data for all effective dates in that policy period’s start date and end date. See “Slice Mode and Window Mode Overview” on page 489.
out-of-sequence	A job issued after another policy change or other job but with an earlier effective date in the same contractual period. If there are conflicts with future-effective-dated branches, users can choose whether to merge changes into future time ranges in that contractual period, or to skip them. Users use the Out-of-Sequence Conflicts tab to merge no changes, some changes, or all changes. See “Out-of-sequence Jobs” on page 501.

Term	Description
preemption	<p>The situation when two concurrent changes are based on the same branch. When the second one finishes, the user must choose whether to apply changes as appropriate from recently-bound jobs into the active job that is about to be bound. Alternatively, the user can withdraw the current job. See "Preempted Jobs" on page 503.</p> <p>A preempted job can also contain out-of-sequence changes. You must handle both issues before binding the job.</p>
merge changes	<p>For an out-of-sequence job, PolicyCenter calculates all out-of-sequence changes that are conflicts. Given these out-of-sequence conflicts, the user can choose to merge those changes in the <b>same branch</b> but at <b>later effective dates</b> in the same policy period. Contrast with the term <i>apply changes</i>. See "Details of Merging and Applying Changes" on page 512.</p>
apply changes	<p>PolicyCenter can calculate all differences between two branches A and B, including entity instance adds, removals, and property changes. PolicyCenter can reapply those differences (the "deltas") to another branch C to recreate what changed between A and B. This occurs as part of handling preemption and processing changes to policies if there is a future renewal. Contrast with the term <i>merge changes</i>. See "Details of Merging and Applying Changes" on page 512.</p>

For a full reference of revisioning properties on the **Policy**, **PolicyPeriod**, and **PolicyPeriod** subobjects, see the next section, "Revisioning Properties Reference" on page 509.

## Revisioning Properties Reference

### Revisioning Properties on a Policy

The policy entity **Policy** is an important entity within PolicyCenter. A policy conceptually serves as a container of contractual periods. From a data model perspective, a policy serves as a container for all versions of its **PolicyPeriod** entities, which is a container for its revisioned subobjects.

The following table lists important revisioning properties on a **Policy** entity instance:

Property	Type	Description
Periods	PolicyPeriod[]	<p>An array of all <b>PolicyPeriod</b> entities associated with this policy including:</p> <ul style="list-style-type: none"> <li>• All contractual policy periods (including renewals, both bound and unbound)</li> <li>• All bound enforced branches</li> <li>• All bound historical (superseded) branches</li> <li>• All draft branches</li> </ul> <p>You typically do not access this property directly. This property contains much data that must be filtered in typical use. Instead, use the <b>BoundPeriods</b> property.</p>
BoundPeriods	PolicyPeriod[]	<p>An array of all bound <b>PolicyPeriod</b> entities associated with this policy including:</p> <ul style="list-style-type: none"> <li>• All contractual policy periods</li> <li>• All bound branches, even superseded ones</li> </ul> <p>You typically do not access this property directly because it contains too much data that must be filtered in typical use. Instead, query to find only the branch you need, such as only the most recent bound branch.</p>

### Revisioning Properties on a Policy Period

A policy revision at one point in model time is represented by the **PolicyPeriod** entity instance. It is the root of a graph of revisioned entities, collectively the **PolicyPeriod** and its subobjects are referred to as a *branch*.

The following table lists important revisioning-related properties:

Property	Type	Meaning
PeriodStart	Date	Date the branch becomes effective. All entities within the branch's graph must have effective and expiration dates on or after this date.
PeriodEnd	Date	Date the period expires. All entities within the branch's graph must have effective and expiration dates on or before this date.
SliceDate	Date	The slice date is the current view, or slice, of the branch (and its entities) in effective time. If the slice date is null then the branch is being viewed/edited in window mode. Any edits made with the slice date set are made in that effective time, splitting the entity instance if necessary. For more information, see "Slice Mode and Window Mode Overview" on page 489. This is a read-only property. To get this PolicyPeriod at a different slice date, use the <code>getAsOf</code> method, described further in "Slice Mode APIs" on page 490.
<b>NOTE:</b> PolicyCenter center never persists the slice date value itself into the database with an object. The slice date property exists as a special property on the in-memory entity instance that Gosu can access.		
Slice	Boolean	If true, this PolicyPeriod is in slice mode (see <code>SliceDate</code> ). Effectively, this is a shortcut to check if <code>SliceDate</code> is non-null. This is a read-only property. To get this PolicyPeriod at a different slice date, use the <code>getAsOf</code> method, described further in "Slice Mode APIs" on page 490.
Promoted	Boolean	If true, this PolicyPeriod was bound, although it is not necessarily the most recent promoted branch for that contractual period. The enforced PolicyPeriod is the one with the highest model number among ones with the same PeriodID. You cannot edit a promoted branch. You must create a new un-promoted branch from a promoted branch and edit it. Until it is promoted, the PolicyPeriod represents an in-progress workspace for a job. This is a read-only property.
ModelDate	Date	On promotion, the model date is set to the current real world date and time. This is a read-only property.
ModelNumber	Integer	On promotion, a branch is assigned a new model number, one greater than the previously most recently promoted branch on its period. This is a read-only property. In contrast, term number, starts with 1 and increments by 1 only for renewals or rewrites.
TermNumber	Integer	The number indicates the term of the policy period, starts with 1 and then increments by 1 for every renewal or rewrite. The built-in PolicyCenter integration with BillingCenter uses the term number instead of the model number to identify a policy.
MostRecentModel	Boolean	Indicates flag that this PolicyPeriod is the most recently bound branch for this contractual period. When a branch binds, if another branch exists in that contractual period: <ul style="list-style-type: none"> <li>• PolicyCenter sets this to false on the previous branch.</li> <li>• PolicyCenter sets this to true on the newest branch in the same database transaction.</li> </ul> Technically, this flag is redundant with checking for the highest model number (ModelNumber) for all PolicyPeriod entities in this contractual period (the same PeriodID). However, use this property to simplify queries that work only with the latest bound branch in any given period. This is a read-only property.
PeriodID	Integer	All branches in the same period share the same PeriodID. This is a read-only property.

Property	Type	Meaning
BasedOn	Integer	The branch ID of the branch this revision was based on. For a standard policy change or renewal, this value is straightforward. However, if a job was preempted by an earlier bound job and it is handled, PolicyCenter creates a new branch. Next, PolicyCenter sets the BasedOn property on the new branch as appropriate. The new value reflects the revised branch ordering after applying changes. For related information, see “Preempted Jobs” on page 503. This is a read-only property.
Id	Integer	This is the Id property present in all Guidewire entities. It is notable because PolicyPeriod subobjects reference this PolicyPeriod by their BranchValue property, which is a cross-reference to this PolicyPeriod property. This is a read-only property.  <b>IMPORTANT:</b> From Gosu, the foreign key property appears as the BranchValue property, although the database column name is BranchID. If you use the query builder APIs, specify this property as BranchValue. For more information, see the BranchValue row in the table in “Revisioning Properties on PolicyPeriod Subobjects” on page 511
Locked	Boolean	Indicates that a PolicyPeriod cannot be modified, either because the branch is bound, withdrawn, or discarded. This is enforced at the application level for the PolicyPeriod and all its subobjects. This is a read-only property.

## Revisioning Properties on PolicyPeriod Subobjects

Each PolicyPeriod entity instance represents the root of a graph of revised entities at a specific model time. Each subobject contains various properties linking the object to:

- its containing PolicyPeriod entity instance
- multiple versions of this object across branches
- multiple versions of this object in the same branch across effective time.

Each PolicyPeriod subobject belongs to one and only one PolicyPeriod and every subobject must have a non-null FixedID and BranchValue property.

The following table lists important revisioning-related properties on PolicyPeriod subobjects:

Property	Type	Meaning
EffectiveDate	Date	Date the entity instance becomes effective. If null, it is implicitly the PeriodStart of its branch.
ExpirationDate	Date	Date the entity instance expires (is no longer effective). If null, it is implicitly the PeriodEnd of its branch.
FixedID	Integer	Identifies a single object across contractual policy periods, both: <ul style="list-style-type: none"> <li>• Within the same PolicyPeriod but with different effective dates</li> <li>• Across multiple PolicyPeriod entities in one contractual policy period</li> </ul> This value must be non-null.

Property	Type	Meaning
BranchValue  <b>Note:</b> At the database layer, this is the BranchID column.	Integer	<p>Foreign key to the PolicyPeriod entity instance that contains this entity instance. Within the same branch, all entities must share the same BranchValue value. This value must be non-null. This is a read-only property. This property is a cross-reference to the PolicyPeriod property called ID.</p> <p>From Gosu, the foreign key property appears as the BranchValue property, although the database column name for that property is actually BranchID.</p> <p>If you use the query builder APIs, specify this property as BranchValue:</p> <pre>var transactionQuery = Query.make(transactionType) transactionQuery.subselect("BranchValue", CompareIn, periodQuery, "ID")</pre> <p>In contrast, when writing upgrade triggers and version checks, you typically want the actual column name:</p> <pre>update.set(wcLineColumn, policyLineQuery, "FixedID");  policyLineQuery.compare("ParticipatingPlanID", Relop.Equals, update.getColumnRef("FixedID"));  policyLineQuery.compare("BranchID", Relop.Equals, update.getColumnRef("BranchID"));  policyLineQuery.withDistinct(true);</pre> <p><b>NOTE:</b> Do not confuse BranchValue with the separate properties BranchName and BranchNumber. The properties BranchName and BranchNumber track alternate versions of the policy within the same job for the multi-version quoting in submission, policy change, and renewal jobs. Each version in a multi-version quoting job is a different draft branch, differentiated by the BranchName and BranchNumber properties in the user interface. BranchName and BranchNumber are not critical for managing branches.</p>
VersionList	SOURCETYPEVersionList  For example, for a PersonalVehicle entity instance, its VersionList property is of type PersonalVehicleVersionList.	Contains a version list, which allows you to access properties with array data across effective time in window mode. For more details, see “Slice Mode and Window Mode Overview” on page 489.
BasedOn	Integer	The internal ID of the entity instance of this type that this entity instance was based on. For a standard policy change or renewal, this value is straightforward. However, if a job was preempted by an earlier bound job, PolicyCenter creates a new branch based on the most recent bound branch to handle the preemption. PolicyCenter discards the original branch with the original BasedOn value. This reflects the revised branch ordering after applying changes. For related information, see “Preempted Jobs” on page 503 and “Details of Merging and Applying Changes” on page 512. This is a read-only property.

## Details of Merging and Applying Changes

There are two different ways that PolicyCenter copies changes from a previous job into another job.

- **Merging changes (OOS)** – For an out-of-sequence job, PolicyCenter calculates all out-of-sequence changes that are conflicts. Given these out-of-sequence conflicts, the user can choose to merge those changes in the same branch but at later effective dates in the same policy period. For more information, see “Out-of-sequence Jobs” on page 501.

- **Applying changes (preemption and future renewals)** – PolicyCenter can calculate all differences between two branches A and B, including entity instance adds, removals, and property changes. PolicyCenter can reapply those differences (the “deltas”) to another branch C to recreate what changed between A and B. This occurs as part of handling preemption and processing changes to policies if there is a future renewal. For more information, see “Preempted Jobs” on page 503 and “Applying Changes to Future Renewals” on page 507.

These are very different processes and it is important to understand their differences.

---

**IMPORTANT** Applying changes for preemption (and future renewals) and merging changes for out-of-sequence jobs work very differently. Carefully read this topic to understand the differences.

---

## Applying Changes Details

This section describes the rules PolicyCenter uses when applying changes for preemption (and future renewals). In this topic, the phrase *the same object* means objects with the same fixed ID (matching `FixedID` property values). The fixed ID is discussed further in “What Is a Policy Revision?” on page 479.

To continue a preempted job PolicyCenter needs to:

1. Create a new branch based on the most recently bound branch. PolicyCenter does this to ensure that no legally-binding changes are lost.
2. Next, PolicyCenter calculates changes between the preempted branch (the one that did not yet bind) and the one it was based on.
3. Finally, PolicyCenter applies those changes to the new branch.

To calculate branch differences, PolicyCenter generates a set of difference item objects (`DiffItem` objects) at a low-level database level. Each difference item represents a change, such as a new object, a deleted object, or a change in a property. PolicyCenter represents these different types of differences with subclasses of `DiffItem`:

- New entities generate a `DiffAdd` object.
- Removed entities generate a `DiffRemove` object.
- Changed properties generate a `DiffProperty` object.
- Window changes (effective/expiration window) changes generate a `DiffWindow` object.

PolicyCenter uses different rules for applying changes for preemption based on each difference item subclass. The rules are similar but slightly different between preemption and applying changes to a future renewal.

The rules cover two aspects of applying the difference item in different cases:

- **Can this difference item subclass apply to the new branch?** Whether the `DiffItem` can apply to the target branch. This essentially attempts to detect whether there is a potential conflict. This corresponds to each `DiffItem` subclass’s method `canApplyDiffToBranch`.
- **How to apply this difference to the branch?** If a difference item can apply to the target branch, how to perform it? This corresponds to each `DiffItem` subclass’s method `applyDiffToBranch`.

The rules for applying changes to a renewal period are similar to the rules that govern handling a preemption. The major difference is that for applying changes to a renewal period, PolicyCenter only cares about applying changes effective at the end of the prior period. Only changes at the end of the period would naturally extend into the future renewal period. If a change in the prior period does not extend to the end of the period, PolicyCenter ignores it and does not consider it a conflict. PolicyCenter handles future renewals based on whether the future renewal is already bound. For information on how PolicyCenter handles the timing of applying changes, see “Applying Changes to Future Renewals” on page 507.

The following table lists the rules for each difference subclass for both preemption and also for applying changes to future renewals:

Type	For preemption, can apply change?	For preemption, how to apply change?	For future renewal, can apply change?	For future renewal, how to apply change?
DiffAdd	Yes.	Add entity instance to the new branch. In very rare cases where the period ranges of the preemption branch is different than the preempted branch, the entire entity instance does not fit into the new range. In this case, it shrinks or expands as necessary.	Yes.	Add entity instance to the new period as it looked at the end of the prior period. In the renewal branch it adds for the entire period range. Scalable properties adjust accordingly. If the entity instance terminates in the prior period before the period end, it does not add to the renewal period.
DiffRemove	Only if the removed entity instance exists in the new branch at the date when it was removed in the preempted branch.	Remove the entity instance on the new branch at the expiration date.	Only if the removed entity instance exists at the start of the renewal period.	Remove the entity instance at the start of the renewal period. This effectively removes it entirely from the new branch.
DiffProperty	For a slice mode edit: • Only if the changed entity instance exists on the pre-emption branch at the effective date of the change.  For a non-slice edit: • Only if the changed entity instance exists on the pre-emption branch at the effective date of the change and two entities effective ranges match. This means that the entity instance has not sliced differently in the new branch (a rare case).	Get the entity instance on the preemption branch at the effective date of the change and set the property.	If the changed entity instance exists at the start of the renewal period.	Apply the property change only if the change in the prior period is effective through the end of the prior period's range. If it is, then set the property at the start of the renewal period. Otherwise, the change is ignored.
DiffWidow	No. It is always a conflict.	Not applicable.	No. It is always a conflict.	Not applicable.

## Merging Changes Details

This section describes the rules PolicyCenter uses when merging changes for out-of-sequence jobs. In this topic, the phrase *the same object* means objects with the same fixed ID (matching FixedID property values). The fixed ID is discussed further in “What Is a Policy Revision?” on page 479.

### General rules for merging changes are:

- If an out-of-sequence job adds an object, later effective date slices include the addition.
- If an out-of-sequence job removes an object, later effective date slices include the addition.

- If an out-of-sequence job changes a property on an object, later effective date slices include the addition only if the later effective date slices do not change that property. When the same property on the same object is changed in both branches, if the values are different, this is an out-of-sequence conflict. The user must review changes in the out-of-sequence conflicts tab before binding the job. See “Out-of-sequence Jobs” on page 501 for details.

**Note:** Many types of changes in an out-of-sequence job are not out-of-sequence conflicts. All entity instance add or removals are not conflicts since they always override the later effective date slices. Even property changes are not conflicts if later slices do not modify that property.

- If merging changes causes validation errors, users must choose new values for properties manually. See “Validation Issues and Out-of-sequence Jobs” on page 503 for related discussion.

The following table lists many examples of how PolicyCenter merges changes for out-of-sequence jobs. In the table, Merged Result means, *If you look at the policy on the later effective date, this is the result after PolicyCenter merges changes.*

Out-of-sequence change	Later effective date slice changes	Merged result
Car 3 is added to the policy	Car 2 is removed from the policy	Car 3 remains. Car 2 is removed. Car 3 is still numbered “3”.
Car 3 is added, garaged in CA	Auto Liability limit for CA is increased from 15/30/15 to 100/300/50 and the changes are applied (automatically) to Cars 1 and 2 (because it is a jurisdiction-level coverage).	Car 3 exists and has the new, higher limit.
Auto Liability limit for CA is increased from 15/30/15 to 100/300/50 and the changes are applied (automatically) to Cars 1 and 2 (because it is a jurisdiction-level coverage).	Car 3 is added, garaged in CA. It initially has the lower liability limit in effect as of the effective date for the earlier change.	Car 3 exists and has the new, higher limit for the jurisdiction-level coverage.
Car 3 is added with Collision deductible set to the then-standard 500.	The Collision coverage deductibles are changed from 500 to 250 for all the cars on the policy, which does not include Car 3 at this time.	Car 3's deductible is not changed. The other cars have the lower deductible.
The Collision coverage deductibles are changed from 500 to 250 for all the cars on the policy (which does not include Car 3 at this time).	Car 3 is added with Collision deductible set to the then-standard 500.	Car 3's deductible is not changed. The other cars have the lower deductible.
Car 1 is removed from the policy.	The Collision deductible for Car 1 is changed from 500 to 250.	Car 1 is removed from the policy. It is as if the limit change never occurred because those coverages were already gone by effective date of the later effective date change.
The CA car is removed, causing all jurisdiction-level coverages tied to Vehicles to be removed.	The Auto Liability limit for CA is changed from 15/30/15 to 100/300/50. The policy has only 1 car in CA.	The CA car and CA jurisdiction-level coverages are removed. It is as if the limit change never occurred because those coverages were already gone by effective date of the later effective date change.
The Basic PIP coverage limit for a vehicle is changed from 10k to 20k.	The vehicle is changed from type Private Passenger to Special because it is actually a dune buggy. PIP coverage no longer applies, so it is removed.	The car is type Special and PIP coverage is gone, so the limit change no longer applies.

Out-of-sequence change	Later effective date slice changes	Merged result
The garage location for all the vehicles on the policy is changed from CA to the insured's new address in AZ. This also requires switching the Collision deductible from 250 to 500 (because 250 is not available in AZ, for example).	A new car is added with deductible 250, like the others.	This actually depends on whether the garage location was edited or a new location added.  If a new location is added, the new vehicle would still be listed at the old location.  If a garage location was edited, the new car is now in a new jurisdiction and its deductible is no longer available. The user must select a valid choice. However, PolicyCenter does not support editing the Jurisdiction of a location. If the garaging location changes, instead add a new location for the new garaging location and remove the old one.
A WC class code (exposure) is added.	The Employer Liability Limit (line-level coverage) is increased.	The exposure is tied to the new, higher limit.
The Employer Liability Limit (a line-level coverage) is increased.	A WC class code (exposure) is added.	The exposure is tied to the new, higher limit.
A WC class code (exposure) is edited to increase the amount from 100k to 120k.	The class code description property on the same exposure is edited to adjust the description.	The exposure has the new description and amount is 120k.
The class code description property on a WC exposure is edited to adjust the description to "Description at Time A".	The class code description property on a WC exposure is edited to adjust the description to "Description at Time B".	This is a merge conflict, and the result could be either value. This is handled by the OOS Conflicts tab.
A BOP policy was written with coverage form = Basic (property on PolicyLine). It is now changed to Broad.	Coverage form is changed to Special.	This is a merge conflict, and the result could be either value. This is handled by the OOS Conflicts tab.
Building added to a property policy	Policy level discount added	Discount and new building on the policy.
A car (garaged in CA) is changed from "Special" to "Private Passenger".	The car's garage location is changed to KY, which is a PIP jurisdiction. As of time B, the car is Special, so PIP coverage does not apply.	The car is now Private Passenger, which means that new coverages would apply, such as Basic PIP. These new coverages must be added if they are available and standard. However, a user would need to select values for the coverage terms. If available but not standard, the user must elect where to accept the coverage.

## Policy Differences Between Revisions

Sometimes PolicyCenter needs to compare two policy branches, which are snapshots of a policy at a specific model time for a specific contractual period. You can customize the appearance of these differences and also customize the underlying mechanism to calculate the differences between two branches.

For more information on policy difference configuration, see “Policy Difference Customization” on page 485 in the *Integration Guide*.



## chapter 45

# Multicurrency Policies

Through multicurrency, PolicyCenter provides the ability to write policies that provide insurance for assets in different currencies. For example, a carrier offers a commercial property policy that can include properties in more than one country. With multicurrency, the policy values the assets in the currency of the asset location.

In the base configuration, PolicyCenter provides infrastructure that enables you to create multicurrency lines of business. PolicyCenter operates in both single and multiple currency display modes. Whether you are running PolicyCenter with a single currency or with multiple currencies, PolicyCenter objects and user interface elements for displaying monetary amounts include both an amount and currency property. Objects, such as accounts and contacts, contain properties for specifying currency properties such as a preferred currency.

PolicyCenter is always configured as a multicurrency system, even if only one currency is defined. The data model and the business logic do not change when multicurrency display is set to single currency (the default). The multicurrency user interface elements are only visible in the base configuration when you enable multicurrency display mode. Even when multicurrency display is set to single, the currency-related fields are still populated within the data model. When multicurrency display mode is single, PolicyCenter does not display the user interface elements that enable you to change those fields.

This topic includes:

- “Multicurrency Overview” on page 517
- “Multicurrency Object Model” on page 522
- “Multicurrency User Interface” on page 524

**See also**

- “Configuring Multicurrency” on page 535 in the *Configuration Guide*
- “Configuring Currencies” on page 55 in the *Globalization Guide*

## Multicurrency Overview

PolicyCenter provides support for one or more currencies in a single policy and within a single account. This topic provides an overview of multicurrency in PolicyCenter.

- “Multicurrency Terminology” on page 518
- “Types of Multicurrency Policies” on page 518
- “Exchange Rate for Multicurrency Policies” on page 519
- “Multicurrency and Rating” on page 519

- “Multicurrency and Reinsurance” on page 521

## Multicurrency Terminology

The following terms are associated with multicurrency.

Term	Description
Coverage currency	The currency for a particular coverage term in the policy contract.
Settlement currency	The currency for premium, taxes, fees, and other similar charges in the policy contract.
Exchange rate	The rate at which one currency is exchanged for another.

## Types of Multicurrency Policies

This topic describes some of the types of multicurrency policies that you can configure in PolicyCenter.

### Single Currency Policy

Many single-country carriers write policies in a single currency and accept payments in that currency. In a single currency system, you use a single currency for rates, premiums, and other monetary aspects of the policy. The user is not aware that PolicyCenter supports multiple currencies. However, when you configure a single currency implementation, be aware that monetary objects are designed to handle more than one currency. These monetary object appear in the data model, the product model, Gosu code, PCF files, and APIs.

For information about how to configure PolicyCenter as a single currency system, see “Configuring PolicyCenter for a Single Currency” on page 535 in the *Configuration Guide*.

### Single Currency Policy with Choice of Currency

PolicyCenter provides support for single currency policies in which the user can choose the currency for each policy.

For example, a carrier in the United States writes homeowners policies to insure assets in North America. One customer has properties in the United States and Canada. The carrier creates two separate policies in this customer’s account. One policy insures the United States property in U.S. dollars. The other policy insures the Canadian property in Canadian dollars.

### Policy with Coverages in Different Currencies

Some carriers with multinational clients need to value coverage terms in more than one currency within a single policy. For example, a carrier provides coverages for the insured’s London office with limits and deductibles in British pounds. In the Berlin office, coverage limits and deductibles are in euros. The premium and other amounts due are in British pounds. In PolicyCenter, you can have a single policy that includes coverages in different currencies. In the base configuration, each policy can have only one settlement currency.

You can modify a line of business, such as Commercial Property, to include policies with coverages in different currencies. In the base configuration, PolicyCenter sets the currency for a coverage to the preferred coverage currency of the coverable. Through configuration, you can modify PolicyCenter so that coverages on a coverable can have different currencies.

### See also

- “Creating Multicurrency Policy Lines” on page 536 in the *Configuration Guide*
- “Configuring the Coverage Currency” on page 538 in the *Configuration Guide*

## Exchange Rate for Multicurrency Policies

The *exchange rate* is the rate at which one currency is exchanged for another. In general, the exchange rate is from one currency to another and often not reciprocal in the other direction. For example, an exchange rate market has one exchange rate for converting euros to U.S. dollars and another for converting U.S. dollars to euros. Currency exchange markets, such as the London Market, set exchange rates and fees. Exchange rates vary with time.

Each carrier will have their own requirements such as how often to refresh the exchange rate or which market rates to use. Therefore, the base implementation of PolicyCenter includes an example exchange rate service which demonstrates rating with multicurrency policies. PolicyCenter obtains the exchange rates from a static table which is not updated. Guidewire expects that each carrier will implement their own exchange rate service.

### See also

- “Implementing an Exchange Rate Service” on page 545 in the *Configuration Guide*

## Multicurrency and Rating

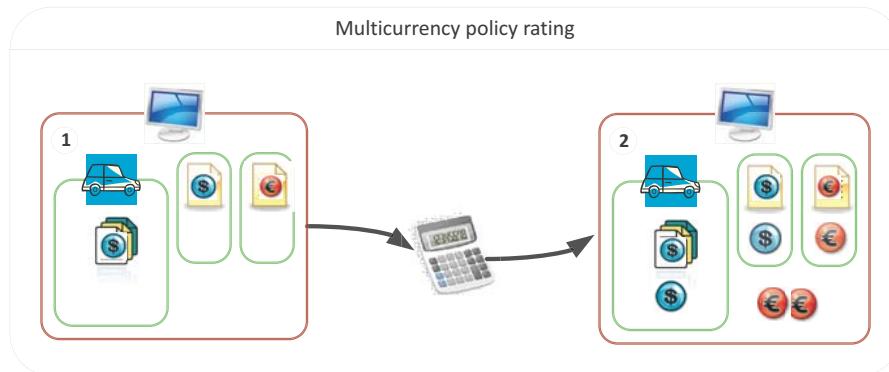
In the base configuration, the rating engine creates cost data in the currency of the associated coverage. For cost data not associated with a coverage, including certain taxes, the rating engine uses the preferred settlement currency on the policy period. The rating engine returns the cost data to PolicyCenter. PolicyCenter creates costs from the cost data, populating the as-rated costs. PolicyCenter then converts the cost amounts to the settlement currency, populating the billing costs, and also saves a pointer to the exchange rate. PolicyCenter stores the settlement currency amounts on the costs. PolicyCenter uses these billing costs to calculate amounts to display on the **Quote** screen, for example, and to present a total cost for the policy.

For billing, PolicyCenter takes the as-rated amounts and computes transactions with amounts in the as-rated currency. Transactions represent changes in cost for billing. Using the same exchange rate, PolicyCenter converts the as-rated currency to the settlement currency and stores the amounts on the transaction. As with costs, the transaction has a pointer to the exchange rate.

### Example

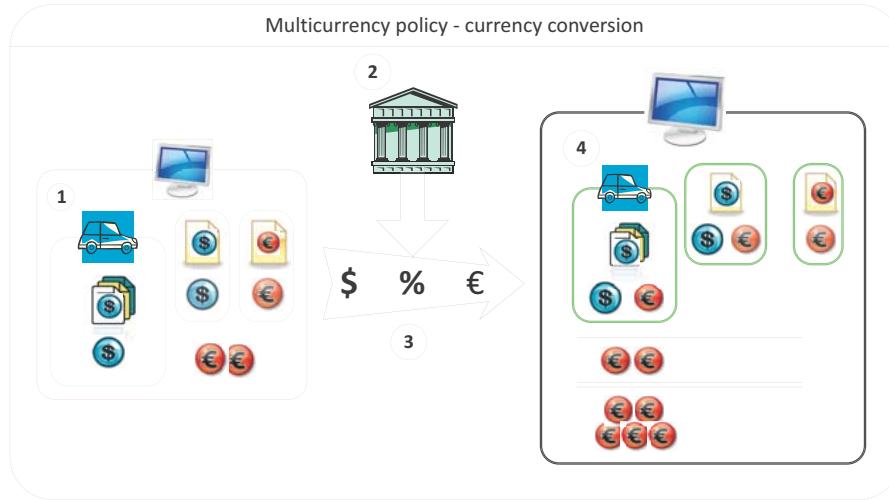
PolicyCenter maintains contractual data such as coverage terms, premium, taxes, and fees in the currency of the policy contract. In a multicurrency policy, rating may require a conversion from one currency to another.

The following example shows rating and currency conversion in a multicurrency policy.



1. In PolicyCenter, a policy has coverables with coverages in different currencies. (In the default configuration, all coverages on a coverable are in the same currency.) Two coverables have a coverage currency in U.S. dollars. One coverable has a coverage currency in euros.

2. The rating engine rates the policy in the coverage currency for each coverage. The rating engine calculates taxes and surcharges in the settlement currency on the policy period.



1. The rating engine has rated the policy in the coverage currency for each coverage.
2. PolicyCenter obtains the exchange rate from the Exchange Rate Service.
3. PolicyCenter converts the as-rated coverage amounts to the settlement currency.
4. PolicyCenter displays the policy premium, taxes, and surcharges in the settlement currency for the policy period. Using the same exchange rate, PolicyCenter creates billing transactions, and converts the amounts to the settlement currency, if necessary.

The policy stores:

- Costs with the as-rated amounts the coverage currency and amounts in the settlement currency.
- The exchange rate
- Billing transactions with amounts in the coverage currency and settlement currency.

#### See also

- “Configuring Multicurrency and Rating” on page 542 in the *Configuration Guide*
- “Calculating Transactions” on page 434

#### [Exchange Rate and Mid-term Change Policy Transaction](#)

In a multicurrency system, a policy transaction which changes the policy in mid-term may or may not result in a change in premium. Policy transactions which change the policy in mid-term are policy changes, cancellations, and reinstatements.

In the default configuration, if a mid-term change policy transaction does not change the cost of a coverage in the as-rated currency, PolicyCenter does not generate a transaction. The policy premium is not affected. If, however, the cost of a coverage does change, then PolicyCenter gets the current exchange rate and creates a transaction representing the change in cost. PolicyCenter converts the as-rated currency to the settlement currency and stores both amounts on the transaction.

The following example shows the transactions that PolicyCenter generates for a coverage in a mid-term policy change. For simplicity, the example assumes that the coverage premium is not prorated from the date of the change.

### Example: Change to Coverable Does not Affect Cost

A customer has a multicurrency commercial property policy. Location coverables can have a coverage currency in euros or British pounds (GBP). The settlement currency is euros. A month into the policy period, the customer calls to report that the name of one of their British locations has changed from *The Meadows* to *The Villages*. This type of change does not affect the cost of the coverable. When the policy was bound, the exchange rate from British pounds to euros was 1.2. Now the exchange rate is 1.1. The customer service representative starts a policy change transaction. In the base configuration, PolicyCenter rates all the coverages. Because this policy change does not affect the cost of the coverable in the as-rated currency, PolicyCenter does not generate a transaction. The following table shows the transaction that PolicyCenter generates for this policy change.

Policy Transaction	As-rated Amount (GBP)	Exchange rate	Billing amount (euro)	Transaction billing amount (euro)
Submission	£1000	1.2	€1200	€1200
Policy change	£1000	1.1	€1100	No transaction

### Example: Change to Coverable Affects Cost

Two months into the policy period, the customer in the previous example calls to increase the coverage limit on one of the buildings at *The Villages*. This change affects the cost of the coverable. When the policy was bound, the exchange rate from British pounds (GBP) to euros was 1.2. Now the exchange rate is 1.3. For simplicity, the example assumes that the coverage premium is not prorated from the date of the change. PolicyCenter re-rates the cost of the coverable. PolicyCenter generates a transaction using the new cost and new exchange rate. The following table shows the transaction that PolicyCenter generates for this policy change.

Policy Transaction	As-rated Amount (GBP)	Exchange rate	Billing amount (euro)	Transaction billing amount (euro)
Submission	£1000	1.2	€1200	€1200
Policy change	£1200	1.3	€1560	€360

#### See also

- “Configuring Multicurrency and Rating” on page 542 in the *Configuration Guide*

## Multicurrency and Reinsurance

The base configuration of PolicyCenter with the Reinsurance Management enabled allows you to manage reinsurance in one or more currencies. The currencies may differ from the currency of the underlying coverages as well as from the settlement currency of the policy. This topic describes the base implementation, which you may extend to handle other reinsurance arrangements.

In PolicyCenter, the reinsurance currency applies to the risks in a policy. The reinsurance currency is independent of the coverage and settlement currencies. The reinsurance currency can be specified independently for each reinsurable risk, based on the characteristics of that risk or other business rules.

In PolicyCenter, a reinsurance program has a specified currency. All agreements in the reinsurance program must have the same currency as the program.

If you have reinsurance programs that covers jurisdictions with more than one currency, then you can create duplicate reinsurance programs for each currency. For example, a carrier has negotiated a set of reinsurance agreements for Europe in euros. The carrier has also negotiated exchange rates to other European currencies. For reinsurance risks in France, the reinsurance currency is euros. For reinsurance risks in England, the reinsurance currency is British pounds. When putting these reinsurance agreements into PolicyCenter, the carrier creates one reinsurance program with the agreements in euros. The carrier creates another reinsurance program with the agreements in British pounds. The carrier manually converts euros to British pounds using the agreed upon exchange rate and enters the converted values in the agreements in the British reinsurance program.

PolicyCenter only attaches a reinsurance agreement to a risk in a policy if the *total insured value/sum insured* (TIV/SI) for that risk has the same currency as the agreement. In the base configuration, the TIV/SI currency is the currency associated with the jurisdiction of the risk. Through configuration, you can modify how PolicyCenter chooses the TIV/SI currency.

In the base configuration, individual risks are aggregated into reinsurable risks based on the location. Every coverable has an associated jurisdiction, so, at least within the base configuration, there are no reinsurable risks that span jurisdictions.

**See also**

- “Reinsurance Management Concepts” on page 589
- “Configuring Multicurrency and Reinsurance” on page 541 in the *Configuration Guide*

## Ceding Premium in a Multicurrency Policy

The carrier cedes a portion of the policy premium to the reinsurer. In many cases, multicurrency policies do not alter the calculations for ceded premium. For proportional agreements, the carrier cedes a percentage of the premium to the reinsurer. For each agreement, PolicyCenter calculates the premium ceded by using the percentage specified in the agreement. This is a simple calculation because all agreements in a program have the same currency; PolicyCenter does not do a currency conversion. For example, an agreement specifies that the carrier cedes 10% of the premium to the reinsurer. If the currency is euros and the premium is 100€, the carrier cedes 10€ to the reinsurer. If the currency is U.S. dollars and the premium is \$100, the carrier cedes \$10 to the reinsurer. A multicurrency policy system does not affect this calculation.

### Ceding a Fixed Amount of Premium

The calculation differs for agreements that cede a fixed amount of premium. A facultative excess of loss agreement, with a fixed cost, is an example of this type of agreement. In this case, the premium ceded is a fixed amount. If the currency of the agreement is in a different currency than the currency of the premium, then a plugin determines how to convert to the premium currency. In the base configuration, the conversion uses the same static conversion table as for other currency conversions. You can configure this to meet your business needs.

For example, a carrier provides multicurrency policies that can have risks in different parts of the world. For risks in Europe, the carrier has negotiated a reinsurance program in euros. This reinsurance program contains a facultative excess of loss agreement that covers losses greater than 100,000,000€. The premium ceded is 10,000€.

This agreement is attached to a policy with a premium of \$30,000 U.S. dollars. This policy covers building in France. To calculate the gross net premium, 10,000€ is subtracted from \$30,000. The plugin calculates how to convert 10,000€ to the premium currency.

The gross net premium is divided among the proportional agreements.

**See also**

- “Calculating Ceded Premiums” on page 608

## Multicurrency Object Model

This topic describes multicurrency object models.

## Multicurrency Properties

The following table shows some multicurrency properties on objects.

Property	Objects	Property description
PreferredCoverageCurrency	Account	The preferred currency for coverages. This is a type key to Currency. Some policy lines are coverables, therefore they also have this property. The Commercial Property line, CommercialPropertyLine entity provides an example.
	PolicyPeriod	
	Coverable	
	Coverable delegates	
PreferredSettlementCurrency	Account Contact PolicyPeriod	The preferred currency for settlement or billing.
CoverageCurrency	Cost	The coverage currency.
	Reinsurable	
	Transaction	
SettlementCurrency	Cost	The settlement currency.
	Transaction	
	WorkersCompLine	

## Multicurrency in a Policy Line

This topic describes objects in a multicurrency policy line using Commercial Property as an example. PolicyCenter displays the user interface for changing the various currencies when the MultiCurrencyDisplayMode parameter is enabled.

---

**IMPORTANT** The MultiCurrencyDisplayMode parameter setting is permanent. After you enable MultiCurrencyDisplayMode by setting the value to MULTIPLE, then start the server, you cannot change the value again. You specify MultiCurrencyDisplayMode in config.xml.

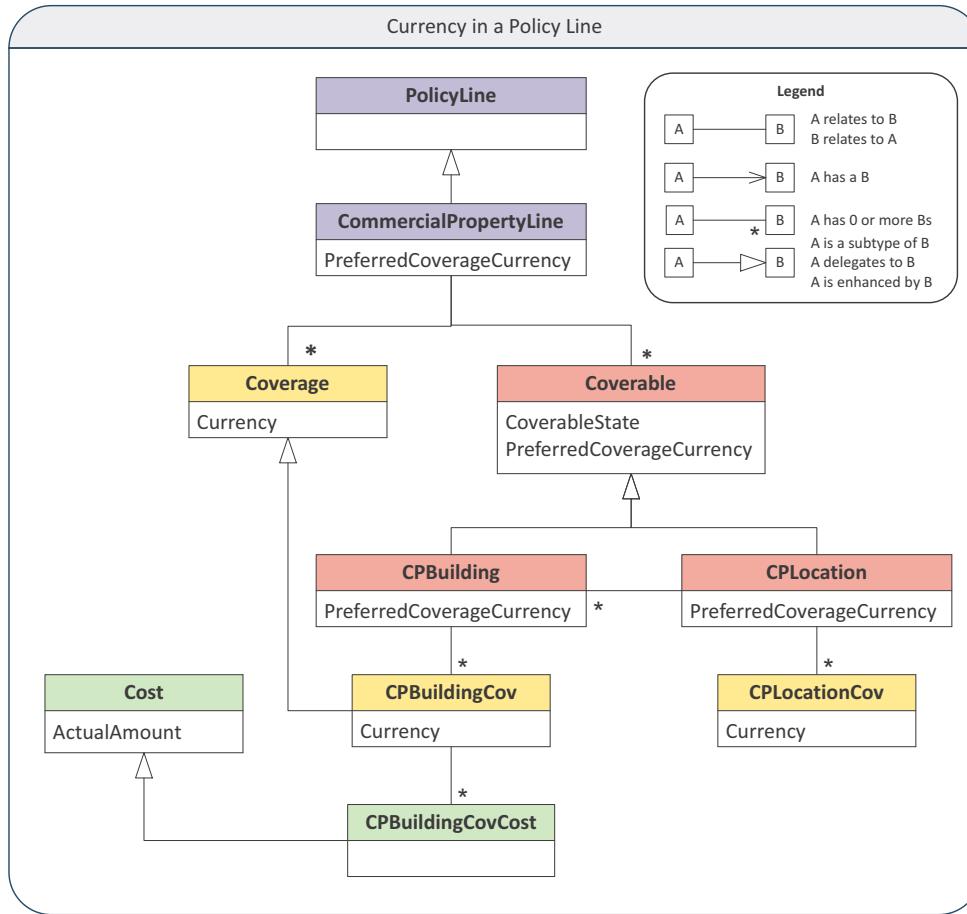
---

In the base configuration, all coverages on the same coverable must have the same currency. Because of this, changing the currency on a specific coverage or clause is not exposed in the user interface. You change the currency on the coverable, and PolicyCenter propagates the currency to the coverages and clauses.

In a line of business, the coverable objects have a preferred coverage currency. You can set this property in the user interface. In the base configuration, PolicyCenter propagates the currency down to the individual clauses (coverages, exclusions, and conditions) for that coverage. In the Commercial Property line of business, the CommercialPropertyLine, CPLocation, and CPBuilding coverable objects have a PreferredCoverageCurrency property. The CommercialPropertyLine is a coverable object, although this object has no line level coverage in the base configuration.

The coverage, exclusion, and policy condition entities (Coverage, Exclusion, and PolicyCondition) include a currency property (Currency) which stores the currency for the clause. The currency property stores the legally binding currency associated with the coverage, exclusion, or policy condition. In the base configuration, PolicyCenter propagates the currency from the coverable; you can modify this behavior through configuration.

In the base configuration, the `CPLocationCov` and `CommercialPropertyCov` coverages have a `Currency` property. PolicyCenter propagates the currency from the coverable.



Both the `PreferredCoverageCurrency` and `Currency` properties have a type key to the `Currency` typelist. You can specify typecodes for additional currencies in this typelist.

The cost object (`Cost`) has a property for the actual amount (`ActualAmount`). This property is a `MonetaryAmount` type.

## Multicurrency User Interface

This topic describes multicurrency fields in the user interface.

PolicyCenter is always configured as a multicurrency system, even if only one currency is defined. The data model and the business logic do not change when configured for single currency (the default setting). The multicurrency user interface elements are only visible in the base configuration when you set the `MultiCurrencyDisplayMode` parameter to `MULTIPLE` in `config.xml`. Even when that parameter is set to `SINGLE`, the currency-related fields are still populated within the data model. When multicurrency display mode is single, PolicyCenter does not display the user interface elements that enable you to change those fields.

You do not need to implement all lines in all configured currencies. All lines do not need to display and allow the user to set the currency. For example, a U.S. Workers' Compensation line only makes sense if the limits are in U.S. dollars.

This topic describes fields that the user sees when multicurrency display is enabled.

## Multicurrency Fields on Policy Transactions Screens

This topic describes the multicurrency fields that appear on screens in policy transactions, such as submissions or policy changes.

### Policy Info Screen

When multicurrency display is enabled, the **Policy Info** screen has several fields related to choosing the currency.

Field	Description
Preferred Currency	If PolicyCenter is configured as a multicurrency system, the <b>Policy Info</b> screen displays this label and the following fields related to the preferred currency.
Coverage	The preferred or default currency for coverages on the policy. The <b>Coverage</b> currency choices come from the policy line configuration in Product Designer.  In the base configuration, the default is the preferred coverage currency on the account, <code>Account.PreferredCoverageCurrency</code> .
Settlement	The preferred or default currency for settlement. This is the currency in which premium, taxes, fees, and the like appear on the <b>Quote</b> and other screens. In the base configuration you can select one of the following currencies: <ul style="list-style-type: none"><li>• USD</li><li>• EUR</li><li>• GBP</li><li>• CAD</li><li>• AUD</li><li>• RUB</li><li>• JPY</li></ul> The currency choices are populated from the <b>Currency</b> typelist. The default is the preferred settlement currency on the account, <code>Account.PreferredSettlementCurrency</code> . You can set the preferred settlement currency in the <b>Account File Summary</b> screen <b>Currencies → Settlement</b> field.

In the base configuration, PolicyCenter validates whether the **Coverage** and the **Settlement** currencies have a supported currency conversion. If PolicyCenter cannot convert the currency, the user receives a validation error. The `IFXRatePlugin` interface has a `canConvert` method which returns true if the plugin can convert from the coverage to the settlement currency.

### See also

- “Implementing an Exchange Rate Service” on page 545 in the *Configuration Guide*

### Coverable Screens

When multicurrency display is enabled, you can set the currency on a coverable in the user interface. PolicyCenter then copies this coverage currency to each coverage on the coverable. The main screen for each coverable has the following multicurrency field:

Field	Description
Coverages in	Select the currency for the coverages on this coverable. By default, this value is set to the <b>Policy Info → Preferred Currency → Coverage</b> field. The drop-down list contains the <b>Available Coverage Currencies</b> defined in Product Designer on the main screen of the policy line.

For an example in the base configuration, see the **Buildings and Locations** screen in Commercial Property. The `CPBuilding` entity delegates to `Coverable`.

PolicyCenter stores coverage terms in generic fields and usually displays these using a widget that is not currency-aware. As a result, coverage terms may not display the currency symbols or abbreviation. The **Coverages in** drop-down list also serves as a consistent user interface element to let the user know the currency for coverage terms not displaying the currency.

### Policy Review Screen

The **Policy Review** screen displays the cost for each coverable's coverages in the currency set on the coverable.

For example, a building is a coverable in a commercial property policy. A policy has one building with coverages in U.S. dollars, and another building with coverages in euros. The **Policy Review** screen displays cost information for the first building in U.S. dollars, and in euros for the second building.

### Quote Screen

The **Quote** screen displays all amounts in the settlement currency.

Continuing the example in the Policy Review Screen topic, assume that the settlement currency is Japanese yen. For each building, the **Quote** screen displays the costs converted into Japanese yen, the settlement currency. Other amounts, such as **Total Premium**, **Taxes & Surcharges**, and **Total Cost** are in the settlement currency.

### Payment Screen

The **Payment** screen displays all amounts in the settlement currency.

## Multicurrency Fields on the Contact Screen

If multicurrency display is enabled, the **Contact File Details** screen has a **Preferred Currency** field in the **Address** section. This field displays the preferred settlement currency on the contact (`Contact.PreferredSettlementCurrency`).

When you create a new contact, the **Preferred Currency** updates when you change the **Country** field on **Address**.

## Multicurrency Fields on the Account Screen

If multicurrency display is enabled, the **Account File Summary** screen has two multicurrency fields under the **Currencies** label. The **Settlement** field displays the account's preferred settlement currency (`Account.PreferredSettlementCurrency`). The **Coverage** field displays the account's preferred coverage currency (`Account.PreferredCoverageCurrency`). For either field, you can select one of the currencies configured in the base application.

Some account screens include summary account information which sums up monetary amounts from multiple policies. For example, the **Account → Billing** screen displays **Account Balances** fields such as **Billed Outstanding** and **Past Due** in the preferred settlement currency of the account. These fields provide an agent a snapshot that approximates the balances on the account, rather than the exact amount that the insured owes in a given currency. Therefore, in the base configuration, the amounts are converted without recording the exchange rates. The screen displays these sums in the preferred settlement currency of the account. If the policy amount is in another currency, PolicyCenter uses the current exchange rate to convert the amount to the preferred settlement currency on the account.

## Multicurrency Fields on the Producer Code Screen

On the **Administration → Users & Security → Producer Codes** screen, each **Producer Code** has a **Currency** field if multicurrency display is enabled. Use this field to specify the settlement currency in which the producer can bind policies.

Although the user interface in the base configuration enables you to select only a single currency for each producer code, the data model supports multiple currencies. The producer code entity (`ProducerCode`) has an array of currencies (`ProducerCodeCurrencies`).

## Multicurrency Field in Reinsurance

If both multicurrency display and Reinsurance Management are enabled, the reinsurance screens have the following fields.

On the **Reinsurance → Search Agreements** and **Reinsurance → Search Programs** screens, you can search by **Currency**. This drop-down list displays the currencies configured in the base application.

On the **Treaty** and **Facultative** screens, the **Currency** drop-down list displays the currencies configured in the base application.

On the **Reinsurance Program** screen, you can select the currency for the program by using the **Currency** field. On the **Treaties** tab, the **Currency** column displays the currency for each agreement.

## Multicurrency Fields for Underwriting Authority

PolicyCenter raises underwriting issues based on characteristics of the policy, including ones that may be related to monetary amounts. You can create and manage monetary underwriting issues with values, approvals, and authority limits that include both an amount and a currency. The base configuration and sample data contain underwriting issues that provide an example of how to compare values, approvals, and authority limits in different currencies.

In the base configuration examples, PolicyCenter creates an issue with a value in the currency on the policy. That currency may differ from the currency of the user's authority grant. In the base configuration, the monetary-issue comparator automatically converts the value being tested to the currency of the reference value. The value being tested might be the value of the user's approval. The reference value currency might be the currency of the user's authority grant. In the sample data, the authority grants are in U.S. dollars, but this is not a requirement. Through configuration, you can take a different approach.

### See also

- “Configuring Underwriting Authority and Multicurrency” on page 544 in the *Configuration Guide*
- “Underwriting Authority” on page 411
- “Risk Approval Details” on page 527

### Risk Approval Details

The **Risk Approval Details** screen is defined in the `UVApprovalLV` PCF file.

When multicurrency display is enabled, the **Risk Approval Details** screen has a **Currency** column. When approving a monetary underwriting issue, you can edit the amount, but cannot change the currency. The **Currency** field is preset to the approval value currency on the underwriting issue.

### Authority Profiles

The **Administration → Users & Security → Authority Profiles** screen shows the authority profiles defined in the system. Each authority profile has one or more authority grants. For authority grants that have a monetary comparator type, the **Authority Grants** table contains **Value** and a **Currency** column.



---

part VI

# Rating Management



# Rating Management Concepts

Guidewire Rating Management provides a set of tools to manage and maintain rating in PolicyCenter.

PolicyCenter provides the ability to rate policies internally or by integrating with an external rating engine. Rating Management enhances the internal rating capability of PolicyCenter by providing a set of tools to manage and maintain rate books, rate tables, and rate routines.

---

**IMPORTANT** To determine whether your Guidewire PolicyCenter license agreement includes Guidewire Rating Management, contact your Guidewire sales representative. Rating Management requires an additional license key. For instructions on obtaining and installing this key, contact your Guidewire support representative.

---

This topic includes:

- “Rating Management Overview” on page 531
- “Rate Tables” on page 533
- “Rate Routines” on page 540
- “Parameter Sets” on page 542
- “Rate Books” on page 543
- “Rating Worksheets” on page 548
- “Impact Testing” on page 549

## Rating Management Overview

Guidewire Rating Management provides tools that you can use to manage and maintain rate books, rate tables, and rate routines in PolicyCenter.

**Rate tables** contain columns representing parameters used to lookup a value, known as a factor. You can use a factor as a rate factor, class, or tier, or as code for use in further lookups. You can view, create, or edit rate tables in PolicyCenter. For example, you can edit rates or add entries for new coverages and limits. For a particular rate

table definition, you can specify the parameter combinations and resulting rate factors to assign to that combination.

**Rate routines** define the algorithm for calculating the rate for coverages, taxes, and other costs on a policy. You can create, view, and edit rate routines in PolicyCenter. The rate routine can reference Gosu methods which:

- Perform utility functions such as polynomial calculations
- Call out to third-party systems to get factors
- Implement complex rating logic

**Rating worksheets** show the actual values that the rate routine used to calculate the rate for a quoted policy or policy transaction. You can use rating worksheets to debug rate routines, validate rates, or get detail showing how a coverage was rated.

**Parameter sets** are associated with rate tables and rate routines. Parameter sets determine the information available for use within a rate routine. Parameter sets also determine the information available for setting the default argument sources in the rate table definition.

**Rate books** define a set of rate tables and rate routines. With a rate book, you can manage rate tables and rate routines as a group, and approve, test, and promote the rate book to production. For example, you can create a rate book that groups a set of rate tables and rate routines that provide the premium for a particular insurance product or offering.

Rating Management provides tools to perform the following types of tasks:

- Change rates or other factors for a given class code, vehicle class, construction type, or territory code.
- Add new entries in a rate table to incorporate new coverages or new limit options.
- Create new rate tables to introduce new factors.
- Create rate routines that define your rating algorithms.
- Test changes to rate books by generating test policy periods with impact testing.

#### See also

- “Quoting and Rating” on page 419
- “Rating Integration” on page 407 in the *Integration Guide*

## Key Features of Rating Management

The key features of Guidewire Rating Management are:

- **Rate table editing** – You can edit and manage rate books, rate tables, and rate routines in PolicyCenter.
- **Microsoft Excel import/export** – Import and export rate tables to Microsoft Excel. With import, you can import existing rate table data into PolicyCenter. With export, you can export rate table data to a spreadsheet and make changes in Microsoft Excel. After making changes, import the changes back into PolicyCenter.
- **Approval and migration** – Provides tools for managing the lifecycle of rate changes, including capturing approvals, managing status, and migrating changes to test and production environments.
- **Rate table versioning and selection** – Availability logic determines which rate book applies for a given product, jurisdiction, effective date, underwriting company, and offering. Rate books contain rate tables and rate routines. Availability does not apply directly to tables and routines. A policy can be rated again midterm with the same set of rate tables and rate routines. At the same time, new versions of those rate tables can be made available for policies sharing the same effective date but bound later.
- **Impact testing** – See the impact that changing the rate book and rate routines has on policy premium. Choose the policies on which to generate test policy periods by product, jurisdiction, producer code, postal code, effective date, expiration date, and as of date. Export the coverage and cost comparisons for each policy to Excel.

- **XML import/export** – Import and export rate books and their associated rate routines from PolicyCenter to XML format. Use import and export to migrate rate books and rate routines between environments.
- **Integration with the product model** – You can link rate table columns or rate routine steps to parts of the product model such as coverages or limits. For example, if you link to a coverage, the product model provides access to the limit options for that coverage.

## Rating Management by Line of Business

Rating management uses a plugin, called the PCRatingPlugin, to rate policies. In the base configuration, this plugin provides rating for the personal auto and commercial property lines of business. You can add rating for other lines of business.

### See also

- “Configuring Rating Management” on page 493 in the *Configuration Guide*
- “Guidewire Rating Management and PCRatingPlugin” on page 411 in the *Integration Guide*

## Sample Data for Rating Management

In the default PolicyCenter application, the sample data includes rate books, rate tables, and rate routines for use with Guidewire Rating Management. You can use these as a starting point. PolicyCenter provides sample data for the personal auto and commercial property lines of business. The rating data is in the small sample data set. The rating data is also in the large data set because that set includes the small data set.

The `gw.sampledata.SampleData` class loads the rating data in the small sample data set by calling the `loadDataSet` method:

```
loadCollection(new SmallSampleRatingData())
```

### See also

- “Installing Sample Data” on page 52

## Rate Tables

Rate tables contain one or more rows with parameter values and the associated return factor value or values, if the rate table has multiple factors. Rate tables are associated with a particular line of business.

## Rate Table Overview

Parameter values can be of any data type. In the default implementation, values can be strings, integers, decimal numbers, boolean values, and dates.

The return values can be:

- Numeric – For factors and rates. Can be decimal or integer.
- Strings – For class definitions, tiers and identifiers.

The following illustration shows a rate table with one factor.

The screenshot shows a table titled "Rate Table Content" with a "Export To Spreadsheet" button. The table has columns: Min (>=), Max (<=), Jurisdiction, and \* Factor. A red box highlights the first row (Min >= 0, Max <= 4500, \* Factor 0.44). Another red box highlights the second row (Min >= 4501, Max <= 6000, \* Factor 0.5). Callout 1 points to the value 4501 in the Min column. Callout 2 points to the value 25000 in the Max column. Callout 3 points to the value 0.5 in the \* Factor column. Callout 4 points to the word "Kentucky" in the Jurisdiction column. Callout 5 points to the value 1.5 in the \* Factor column.

	Min (>=)	Max (<=)	Jurisdiction	* Factor
	0	4500		0.44
	4501	6000		0.5
	6001	8000		0.65
	8001	10000		0.8
	10001	15000		0.9
	15001	20000		1
	20001	25000	Kentucky	1.2
	20001	25000		1.5
	25001	40000		1.5
	40001	65000		2
	65001	90000		3
	90001			4

In the illustration, the numbered items represent:

1. Parameter values for the Min (>=), Max (<=), and Jurisdiction parameters.
2. Factor value – The return value if the row matches. All factors appear to the right of the gray column.
3. Row – Provides parameter values and one or more rate factors.

If the policy data matches parameter values in a row, PolicyCenter returns the factor value. Rate routines associated with this line of business can access these factor values.

## Rate Table Definition

The Rate table definition specifies parameters and the factor required for a given rate table. Each parameter has a matching rule, name and data type, and optional value provider. The matching rules specify how to compare a policy value to the rate table's parameter value. In the base configuration, the matching rules include exact match, range matches, longest substring, less than or equal, greater than or equal, and interpolated match.

### See also

- “Matching Rules” on page 559
- “Configuring Matching Rule Operations” on page 501 in the *Configuration Guide*

## Value Provider

On the **Parameters** tab of a rate table definition, the **Value Provider** specifies a parameter value by selecting from a drop-down list of available values. The list of values comes from data within the data model or data within another rate table. If you do not specify a value provider, the parameter value is a free-form data entry field in the **Rate Table Editor**. The default implementation includes the following value providers:

- **Typelist Value Provider** – Specify any PolicyCenter typelist to limit the possible values for this parameter. Select a typelist value from a drop-down list for this parameter in the **Rate Table Editor**.
- **Product Model Coverage-related Value Providers** – Leverage the coverage, coverage terms, and coverage term option values already specified in PolicyCenter. Select a value from a drop-down list for this parameter in the **Rate Table Editor**. For coverage term and coverage term values, the list is further restricted based on the selected coverage or coverage term.
- **Reference Factor Value Provider** – Specify a factor column in another rate table that provides the possible values for this parameter. Select a value from a drop-down list of values from the reference column in the **Rate Table Editor**.

### Using a Reference Factor Value Provider in Rate Tables

For some rate table information, the factors from one rate table can be parameters to another rate table. Although a single rate table can represent this information, linking two rate tables in this way can make the information easier to maintain. You can use a reference factor value provider to specify a factor column in one rate table that provides values for a parameter in another rate table.

For example, on a personal auto policy, the model year, manufacturer, model, and style map to a class code factor in a rate table. The following table shows a small example of the data.

Year minimum	Year maximum	Manufacturer	Model	Style	* Class code
2001	2004	Toyota	Prius		CCAH
2004		Toyota	Prius		IDAH
2011		Nissan	Leaf		JDAE
1993	1997	Volvo	850	Station Wagon	GWMR
1993	1997	Volvo	850	4-door Sedan	GDAR

In another rate table, the class code and age of the vehicle determine the rate factor. The first table links to this rate table because the class code factor from the first table is a parameter in this table.

Class code	Vehicle age	* Factor
CCAH	1	1.0
CCAH	2	0.96
CCAH	3	0.91
CCAH	4	0.85
CCAH		0.78
IDAH	1	1.1
IDAH	2	1.05
IDAH	3	0.98
IDAH	4	0.89
IDAH		0.78

You can configure additional value providers. See “Configuring Value Providers” on page 499 in the *Configuration Guide* for implementation details of each included value provider and instructions for configuring a new value provider.

### Physical Tables and Entities for Rate Table Definitions

The rate table definitions are logical representations that map to physical tables in the database. An entity in the *Data Dictionary* describes the properties of the physical table. The physical table contains properties to hold the rate table content in rows. When you create a rate table definition, you must specify the physical table. You can use the same physical table to store the contents of multiple rate table definitions.

For example, the default implementation includes a default, generic physical table that many of the rate tables for most lines of business can use. The `DefaultRateFactorRow` entity represents the generic physical table. If the generic physical table does not meet your needs, you can create an entity to represent your own physical table.

The generic physical table is appropriate if:

- The number of rows expected in the rate table is less than the memory threshold. See “Memory Threshold for Physical Tables” on page 536.

- The parameter and factor columns can map to the available data types on the generic physical table. The table contains:
  - Eight string columns
  - Eight integer columns
  - Six decimal columns
  - Two date columns
  - Two boolean columns

## Custom Physical Tables

The default implementation includes one sample custom physical table. This sample custom physical table contains properties for coverages and coverage term values parameters to derive a single decimal factor. This custom physical table, the `CoverageRateFactor` entity, contains the following columns:

- Coverage Code
- Coverage Term Code
- Coverage Term Option Code
- Jurisdiction
- Factor

For rate tables larger than the memory threshold, use a custom physical table to improve performance. Add indexes to the entity for efficient retrieval.

### See also

- “Custom Physical Tables for Rating Management” on page 498 in the *Configuration Guide* for details on creating custom physical tables

## Memory Threshold for Physical Tables

The memory threshold sets the limit below which a rate table is loaded into memory. If a rate table exceeds this threshold, the table is queried directly from the database rather than being loaded into memory. This can affect performance. The memory threshold applies to all physical tables that hold rate tables. In the base configuration, the memory threshold is 1000.

For example, two rate tables use the generic physical table. The first rate table has 500 rows. The second rate table has 900 rows. Another version of this rate table has 950 rows. The number of rows in each version of the first two rate tables is less than the memory threshold. Therefore, you can use the generic physical table for these rate tables.

A third rate table has 2000 rows. The number of rows in this rate is above the memory threshold. To load this rate table from disk into memory, you must increase the memory threshold. If you leave the memory threshold at the current value, the table remains on disk, and querying may be slow.

You can increase the memory threshold to a higher value. However, increasing the threshold can affect system performance. Test the system performance by loading a physical table that contains a number of rows nearing the memory threshold.

To change the memory threshold parameter, edit the `rating.properties` file in `modules/pc/config/resources`. The memory threshold parameter is `memory.threshold`. In the base configuration, this parameter is set to 1000.

## Matching a Factor in the Rate Table

The second stage of the rating query determines the most suitable factor in the rate table and returns the factor to the rating engine.

The query takes a list of input values, corresponding to the parameters relevant to the rate table. The query compares the input values to the rate table rows to determine the best match. The query returns the value of the matching row.

In PolicyCenter, the rate table displays the parameters from left to right in descending priority order. Lower numbers have higher priority, so that column 0 is higher priority than 10. The query looks for a complete match. In a complete match, all parameters must have a matching value. If found, the query returns the factor from the matching row. If not, the query matching requirements are relaxed by removing the lowest priority parameter from the matching criteria. The query matches a row in which the lowest priority parameter is blank and all other parameters match. The relaxing of low priority parameters continues until a match is found or all parameters have been relaxed.

### Matching Range Parameters

A range parameter has two arguments and appears in the rate table as two columns: a lower and an upper limit. A range parameter is blank if both arguments are blank. The whole parameter is relaxed as a single unit.

A blank on one side of a range always matches. For example, a rate table has the following ranges:

Minimum	Maximum	* Factor
	1000	0.50
1000	2000	0.75
2000	5000	1.00
5000		1.25

The matching rule for the parameter is set to exclude the maximum value. The first range matches a value under 1000, and the last range matches a value of 5000 or more.

### Example of Finding a Factor

This example provides a simple rate table definition with two parameters for a salary: jurisdiction and amount. The **Jurisdiction** parameter is an exact match matching rule with priority 1. The **Amount** parameter is a greater than or equal matching rule with priority 10. In the rate table, PolicyCenter displays parameters in descending priority order, therefore **Jurisdiction** appears before **Amount**.

In PolicyCenter, the rate table has a **Jurisdiction**, **Amount**, and **\* Factor** column. The following table has additional **Row** and **Description** columns.

Row #	Jurisdiction (exact)	Amount ( $\geq$ )	* Factor	Description
1			1.00	Matches any salary if both <b>Amount</b> and <b>Jurisdiction</b> are relaxed. This factor is the default factor.
2		80,000	2.50	Never matches because relaxing occurs from higher priority to lower priority, or right to left. Since <b>Amount</b> is specified, all parameters to the left must be specified.
3	CA		1.25	If <b>Amount</b> is relaxed, matches any salary in the California jurisdiction.
4	CA	30,000	1.50	Matches a salary in the amount of \$30,000 or more in the California jurisdiction.
5	CA	50,000	1.75	Matches a salary in the amount of \$50,000 or more in the California jurisdiction.
6	NY		1.50	Matches any salary in the New York jurisdiction when the <b>Amount</b> is relaxed.
7	NY	50,000	1.75	Matches a salary in the amount of \$50,000 or more in the New York jurisdiction.

A person who lives in California and has a \$35,000 salary matches **Jurisdiction** and **Amount** in row 4 and gets a 1.50 factor. A person who lives in California and has a \$20,000 salary does not match any **Jurisdiction** and **Amount** specified for California. Therefore, **Amount** is relaxed. The salary matches row 3 and gets a 1.25 factor.

A person who lives in Florida and has a \$55,000 salary does not match the **Jurisdiction** and **Amount** for any row. When **Amount** is relaxed, the salary still does not match any **Jurisdiction**. When **Jurisdiction** is relaxed, the salary matches row 1.

#### **Example of Finding a Factor with a Range Parameter**

This example is similar to the previous example, but defines salary with a range matching rule rather than a greater than or equal matching rule.

A range parameter for salary appears as two columns in the rate table and has a priority for each column. The two arguments of the range parameter are **Min (>=)** and **Max (<=)**. The **Min (>=)** argument is priority 10, the **Max (<=)** argument is priority 20. PolicyCenter relaxes the range parameter as a single unit.

In PolicyCenter, the rate table has **Min (>=)**, **Max (<=)**, and **\* Factor** columns. The following table has additional **Row** and **Description** columns.

Row #	Jurisdiction (exact)	Min (>=)	Max (<=)	* Factor	Description
1				1.00	Matches any salary when both the salary range parameter and <b>Jurisdiction</b> are relaxed. This factor is the default factor.
2	NY			1.25	Matches any salary in New York when the salary range parameter is relaxed.
3	CA	0	30,000	1.25	Matches any salary in California when the amount is between \$0 and \$30,000.
4	CA	30,001	50,000	1.50	Matches any salary in California when the amount is between \$30,001 and \$50,000.
5	CA	50,001	80,000	1.75	Matches any salary in California when the amount is between \$50,001 and \$80,000.
6	CA	80,001		2.00	Matches any salary in California of \$80,001 or greater.

A person in California who has a \$35,000 salary falls between **Min (>=)** and **Max (<=)** in row 4 and gets a 1.50 factor. A person in California who has a \$90,000 salary matches row 6 and gets a 2.00 factor. A person in New York who has a \$50,000 salary does not match any row. Row 2 matches when salary, **Min (>=)** and **Max (<=)**, is relaxed. A person in Florida who has a \$35,000 matches row 1 after both salary and **Jurisdiction** are relaxed.

#### **Rate Table with Multiple Factors**

A rate table that returns multiple factors returns more than one factor with each table lookup. You can use a multiple factor rate table if you have parameter values associated with more than one factor.

For example, in a building factor rate table, the class code determines the minimum premium and risk factors. The rate table has a **Class Code** parameter, and a **Minimum Premium** and **Risk** factor.

Class code	Minimum premium	Risk
00001	2,000	0.05
00002	4,000	0.01
00003	2,500	0.25

If a rate routine uses this rate table, the routine does a single table lookup on the **Class Code** to access to both the **Minimum Premium** and **Risk factors**. The routine can access each factor separately in the rate routine steps.

You can use a multiple factor rate table to improve performance and maintainability if you have two or more large rate tables with identical parameters. In personal auto, for example, you have a rate table with make, model, year, and jurisdiction parameters. A multiple factor rate table can store a factor for each coverage. To add the coverage factors for a 2013 Toyota Camry, you add one row to the table for each jurisdiction. Your rate routine can fetch all factors in a single query, and access each factor separately in later steps.

In some cases, you may choose to combine rate tables with similar parameters into a multiple factor rate table. Consider whether maintaining this rate table is more difficult than maintaining a multiple factor rate table formed from two rate tables with identical parameters.

### Rate Table with Interpolated Rate Factors

In a rate table with interpolated rate factors, PolicyCenter computes the rate factor if the policy value falls between two interpolated parameter values. A rate table can have only one interpolated parameter.

The data type of the parameter must be a numeric type: either decimal or integer. The data type of the factor must also be numeric. The parameter and the factor can have different numeric data types. For example, a rate table has integer parameters because they are discrete end points of a range, but the factors are decimal values. In this example, the policy values can be decimal rather than integer.

The **Interpolation Param** rate table in the sample data is an example of a rate table with interpolated rate factors. This rate table is for the personal auto line of business.

The following example shows a rate table similar to, but not the same as, the **Interpolation Param** rate table. (The **Row #** column does not appear in PolicyCenter.) The rate table has the following content:

Row #	Jurisdiction	Interpolated parameter	Discount code	Factor
1				14
2	New Jersey			13
3	New Jersey	0		10
4	New Jersey	6		8
5	New Jersey	10		6
6	New Jersey	10	1234	4

If the policy values match row 4 exactly or by relaxing parameters, then PolicyCenter matches row 4 and returns the factor of **8**. For example, the policy values match row 4 if the **Jurisdiction** is **New Jersey** and the **Interpolated Parameter** is **6**. Because the **Discount Code** is not specified, it is disregarded in the match.

The only way you can match row 6 is with the exact inputs: **New Jersey**, **10**, and **1234**.

If the policy values are **New Jersey** and **Interpolated Parameter** of **11**, the interpolated parameter value does not fall between any two rows. Therefore, the **Interpolated Parameter** parameter is relaxed, and the policy values match row 2.

If the policy value of the interpolated parameter falls between two rows and matches on the other parameters, then the factor is calculated proportionally. The interpolated factor is computed according to this formula:

$$\text{IF} = \left( \frac{(\text{PV} - \text{IP1})}{(\text{IP2} - \text{IP1})} \times (\text{F2} - \text{F1}) \right) + \text{F1}$$

Where:

- **IF** – is the interpolated factor

- PV – is the policy value of the interpolated parameter
- IP1 – is the lower value of the interpolated parameter
- IP2 – is the higher value of the interpolated parameter
- F1 – is the value of the factor for IP1
- F2 – is the value of the factor for IP2

For example, the policy values are **Jurisdiction** of **New Jersey**, **Interpolated Parameter** is **8**, and **Discount Code** is **null**. The interpolated factor is computed by using rows 4 and 5 in the rate table. The formula is:

$$\begin{aligned} \text{IF} = & ( \\ & ( (8 - 6) \div (10 - 6) ) \\ & \times (6 - 8) \\ & ) \\ & + 8 \end{aligned}$$

Since the formula is a bit complicated, the following lines show in detail how to compute the input factor:

$$\begin{aligned} \text{IF} &= ( (2 \div 4) \times (6 - 8) ) + 8 \\ \text{IF} &= ( \frac{1}{2} \times (-2) ) + 8 \\ \text{IF} &= (-1) + 8 \\ \text{IF} &= 7 \end{aligned}$$

## Rate Routines

Use rate routines to implement rating algorithms that calculate the properties on the cost for coverages, taxes, and other costs on a policy. The properties on the cost include the base rate, adjusted rate, and term amount. You can also create rate routines that do not calculate these properties. These types of rate routines may set up rating information for use by another rate routine.

Each rate routine has one or more instructions that define the rating algorithm. Each instruction is composed of one or more steps. The steps implement your rating algorithm or provide supporting logic for the algorithm. For example, a step can determine the drivers on a policy or perform a composite calculation.

You define rate routines for a particular line of business. Rate routines at the policy level run after running the rate routines for individual coverages.

For each rate routine, you specify a *parameter set* which enables you to pass contextual information to the rate routines. In rate routines, parameters provide access to policy details related to rating, such as driver age or policy effective date.

The rating engine defines the criteria that PolicyCenter uses to select the rate routine. For example, the rating engine can select a rate routine by using criteria such as the following:

- The line of business.
- A coverage on the policy. The rate routine can also apply based on a characteristic of the covered item. For example, you can specify a general rate routine for cars and a special rate routine for sports cars.

In the base configuration, the small and large sample data contains a set of rate routines for the personal auto and commercial property lines of business. Each rate routine has a code, name, and version. To calculate a rate, the rating engine selects the appropriate rate book edition and rate table version. Then, the rating engine executes the rate routine in that rate book to get a cost.

### See also

- “Selecting the Rate Book Edition During Policy Rating” on page 544
- “Configuring the Rating Engine to Execute the Rate Routine” on page 503 in the *Configuration Guide*

## Rate Routines That Do Not Calculate Properties on the Cost

Most rate routines calculate properties on the cost such as the base rate, adjusted rate, and term amount. You can also create rate routines that do not calculate these properties. These types of rate routines may set up rating information for use by another rate routine.

In the sample data, the **PA Assign Driver** rate routine does not calculate properties on the cost. This rate routine sets the assigned driver which another rate routine uses. The **PA Assign Driver** rate routine accesses the **Youthful Driver Age** rate table to determine whether the driver qualifies as a youthful driver. This rate routine provides data for the **PA Vehicle Coverage Premium Algorithm** rate routine that rates vehicle coverages. The rating engine executes **PA Assign Driver** before executing **PA Vehicle Coverage Premium Algorithm**.

### See also

- “Creating Rate Routines That Do Not Set Properties on the Cost” on page 508 in the *Configuration Guide*

## Rate Routine Versions

Create a new version of a rate routine when you need to modify it. Versions start at the number 1 and increase by 1 for each new version. If a rate book references version 1 of a rate routine, that rate book continues to reference version 1. You can select the new version of the rate routine in a new rate book or by editing an existing rate book.

## Rate Routine Variant Identifiers

In some cases, the rate routine differs in one or more jurisdictions. In the base configuration, you can specify a rate routine that applies to a specific jurisdiction. This is known as a *jurisdiction variant* in the user interface. A rate routine and its jurisdiction variants have the same code.

The jurisdiction variant is an example of a rate routine variant identifier. In the base configuration, jurisdiction variant is the only example of a variant identifier. However, through configuration, you can create variant identifiers to other features of the policy such as the underwriting company. For more information, see “Configuring Variant Identifiers for a Rate Routine” on page 506 in the *Configuration Guide*.

## Rate Routine Steps

Each rate routine has one or more instructions that define the rating algorithm. Each instruction is composed of one or more steps. Each step has the following fields:

- Instruction – This field is the target of an assignment operator, a conditional instruction, or the start of a section comment. Blank for other operators. The instruction field types can be properties on the cost, parameters including rate modifiers, variables, and conditionals. This instruction field is not the same as an instruction, which is composed of one or more steps.
- Operator – An assignment, arithmetic, or rounding operator. Blank for conditional instructions.
- Opening parentheses (optional) – One or more opening parentheses to group a series of steps.
- Operand – The operand of the current step. Operand types can be properties on the cost, functions, parameters, rate tables, variables, conditional expressions, constants, date constants, typelist values, and scale for rounding operators.
- Closing parentheses (optional) – One or more closing parentheses to group a series of steps.
- Line comment– Optional description of the step.

For example, you can represent a rate routine that calculates the coverage premium for personal auto. In the following table, each row corresponds to a step:

#	Instruction	Op	( Operand )	Line comment	
1	--	<b>Base Rate and Adjusted Rate Calculation</b>			
2	BaseRate	←	table:Base Rate	The instruction field is the BaseRate field on a cost. The ← operator assigns the operand to the BaseRate in the instruction field.	
3		R	.001	Round Base Rate to thousandths (Round Half Up)	
4	vehicleTypeFactor	←	table:Vehicle Type Factor		
5	vehicleCostFactor	←	table:Vehicle Cost Factor		
6	AdjustedRate	←	BaseRate	Determine adjusted rate. The instruction field and operand are properties on the cost.	
7		X	table:Coverage Factor	Operand is a rate table.	
8		X	table:UW Co Factor	Operand is a rate table.	
9		X	Policy Line.PAMultiCarDiscount	Operand is a parameter.	
10		X	min(vehicleCostFactor, vehicleTypeFactor)	Operand is a function.	
11					
12	--	<b>Term Amount Determination</b>			
13	IF		AdjustedRate < table:Min Premium AND Vehicle.VehicleType = Vehicle-Type.Other	The instruction field begins a conditional instruction, and operand is a conditional expression.	
14	TermAmount	←	table:Min Premium	The instruction field is the TermAmount field on a cost, and operand is a rate table.	
15		RU	10	Round up minimum premium to tens.	
16	ELSE			Continue the conditional instruction begun in step 13.	
17	TermAmount	←	AdjustedRate	Use system default rounding.	
18	ENDIF			End of the conditional instruction begun in step 13.	

In the rate routine, steps 2 and 3 make up one instruction. Steps 4 and 5 are each a single line instruction. Steps 6 through 10 are one instruction. Steps 13 through 18 are a conditional instruction.

Steps 1 and 12 are section comments that span the whole step.

#### See also

- “Working with Rate Routines” on page 565

## Parameter Sets

In PolicyCenter, you can associate parameter sets with rate tables and rate routines. Parameter sets define the contextual information passed to rate routines. In rate routines, you can use parameters to access policy details related to rating, such whether the car has antilock brakes.

In the rate table definition, you specify the default argument source for each parameter in the parameter set. When you specify a rate table in a rate routine step, the default value is the default argument source. You can override the default value.

The rate table definition can include more than one set of default argument sources. When you specify a rate table in a rate routine step, you can choose the argument source set.

The parameter set must include a policy line parameter. Through the policy line, you can access policy periods and other parts of the policy.

In a parameter set, you specify parameters that the rate routine needs for calculation. You can specify parameters for entities in PolicyCenter, for example, policy line entities, coverages, or objects used in rating such as rating information and rate date.

## Rate Books

Rate books group related rate tables and rate routines. Grouping related rate tables into a rate book eliminates the need to track availability and effective dates at the individual rate table level. As a unit, the rate book can be versioned and promoted to active status for use in production. You cannot modify the rate tables and rate routines in an active rate book.

In rate books, you can define qualifying attributes such as jurisdiction, underwriting company, and offering.

A rate book is identified by a code, name, description and version and has the following availability attributes:

- Policy line – Required
- Underwriting company – Optional, default is “any”
- Jurisdiction – Optional, default is “any”
- Offering – Optional, default is “any”
- Policy effective or coverage reference date – Required
- Activation date – set to the date that the rate book was activated into production

## Managing Rate Books and Rate Tables

Changes to rating can be driven by regulatory changes or business reasons. The majority of rating changes involve changes to the rate table rows of one or more rate tables.

Rate books group related rate tables. More precisely, each rate book edition groups a specific set of rate table versions. When you edit a rate table in a rate book, PolicyCenter creates a new version. The rate book includes the new rate table. You can manage rate book editions in PolicyCenter.

As soon as you activate a rate book, the rate book is available for production use. To ensure the integrity and auditability of policies rated with the activated rate book, you cannot update an active rate book or its contents. If any rate table in an active rate book requires a change, you must create a new edition of the rate book. In the new edition, you can update the rate tables. Updated rate tables are assigned a new version internally, in the context of that new edition of the rate book. The system automatically creates a new version of that rate table. This behavior allows previous versions of the rate table to be preserved in existing rate books that reference or own the specific version of the rate table. To avoid proliferating rate table versions unnecessarily, rate tables that remain unchanged in the newly versioned rate book continue to reference the existing rate table.

For this reason, there are three different relationships between a rate book edition and a rate table version. The relationships are:

- **Referencing** – Another rate book edition owns this rate table version. Updating it results in a new **Owned – Not Shared** rate table version.

- **Owned – Shared** – This rate table version was created in this rate book edition but other rate book editions now reference this same rate table version. This relationship type only applies to active rate books.
- **Owned – Not Shared** – This rate table version was created in this rate book edition but no other rate book editions yet reference this same rate table version. Provided the rate book is still in draft mode, you can update this rate table version without creating a new version of the rate table.

## Selecting the Rate Book Edition During Policy Rating

In Guidewire Rating Management, a rating query selects the appropriate rate book edition and rate table version.

The rating query executes the query in two stages:

1. Determine and retrieve the appropriate rate book based on the coverage and policy attributes and the “Rate Book Matching Process” on page 544. Within the matching rate book, determine the correct rate table version required for the second stage of the query.
2. Retrieve the correct factor from the relevant rate table version, based on the input values provided.

## Rate Book Matching Process

The rating query uses the following attributes to select a list of candidate rate books:

- Required attributes – Attributes that must match for a rate book to be a candidate.
  - Policy Line – The rate book policy line must be the same as the policy’s policy line.
  - Effective/Expiration Dates – The coverage **Reference Date** must be later than or equal to the rate book **Effective Date** and earlier than the rate book **Expiration Date**. For more information about the coverage reference date, see “Setting the Reference Date” on page 90 in the *Product Model Guide*.
  - Activation Date – The rate book **Activation Date** must be earlier than the policy **Rate as of Date**. The rate as of date controls which version of a rate book to use if there are rate book editions with overlapping effective dates.
  - Status – The rate book **Status** is equal to or higher than the **Minimum Rating Level**. The status is a configuration parameter. See “Minimum Rating Level Parameter” on page 495 in the *Configuration Guide*.
- Optional attributes – Attributes used to attempt an exact match with a rate book. Attributes are relaxed (look for a rate book with null/any in that field) one by one until a matching rate book is found.
  - Jurisdiction
  - Underwriting Company
  - Offering
- Relaxing Rules
  - The rate book attributes are relaxed in the following order: Offering, UW Company, Jurisdiction.
  - The process terminates as soon as any rate books that match the relaxed query are found.

The rate book matching process is as follows:

1. The query retrieves a list of rate books that match required attributes.
2. Further filter down the list based on the values of offering, underwriting company and jurisdiction.
3. If a single matching rate book is found, return it.
4. If multiple matching rate books are found, return the rate book with the latest activation date that is earlier than the rate as of date.

### See also

- “Configuring Matching Rule Operations” on page 501 in the *Configuration Guide*

## Overlapping Effective Dates

Rate books can have overlapping effective dates.

**How can overlapping effective dates occur?** Within the rate book's effective period, typing errors in the rate table may be discovered or last-minute regulatory changes may arise. These changes require an update to rate tables in already-activated rate books.

**When there are overlapping effective dates, why does the rate book selection process not automatically select the replacement or most recently activated rate book?** In many cases, regulations require that a carrier retain the same set of rates used at issuance for any subsequent policy changes. And in many cases, regulations only allow a rate change at renewal. As a result, any policies bound prior to the replacement rate book being activated would, in most cases, continue to use the original rate book for policy changes. New business or renewals with effective date later than the replacement rate book effective date applies rates defined in the replacement rate book. In the case of a policy change, the policy rate-as-of date is initially set to the calendar date/time that a policy was initially rated. This behavior ensures that subsequent transactions retrieve the same rate book. The rate-as-of date can be modified to allow flexibility in the rate book to use for the transaction.

For example:

- Rate Book v1 is effective 1/1/2011 and activated 10/15/2010 so that it is available for 1/1/2011 renewals which begin processing 11/1/2010.
- On 11/15/2010, Policy A effective 1/15/2011 is rated and bound. Rate Book v1 is used, and the rate-as-of date is 11/15/2010.
- On 12/1/2010, Rate Book v2 with updated 2011 rates is introduced. Rate Book v2 is also effective 1/1/2011 but activated 12/1/2010.
- On 12/15/2010, Policy B effective 1/15/2011 is rated and bound. Rate Book v2 is used, and the rate-as-of date is 12/15/2010.
- If both Policy A and Policy B were endorsed effective 2/1/2011, Policy A uses Rate Book v1 and Policy B uses Rate Book v2.
- By using the rule referenced above (use rate book with highest activation date before the policy rate-as-of date):
  - Policy A (rate as of date = 11/15/2010) picks up Rate Book v1 (activation date = 10/15/2010) and NOT Rate Book v2 (activation date = 12/1/2010).
  - Policy B (rate-as-of date = 12/15/2010) picks up Rate Book v2 (activation date = 12/1/2010).

The default behavior of policy rate-as-of date is as follows:

- Rate-as-of date is at the policy period level.
- Rate-as-of date is set to the current date/time whenever a new term (submission, renewal, rewrite) is rated and is read-only. At bind/issue, this date represents the last date that the new term was rated.
- For policy changes, the system does not update the rate-as-of date, so the date defaults to the date set at issuance. However, for policy changes, a user can modify the rate-as-of date to deliberately re-rate a particular policy change (or in bulk) with a new rate book. The user must have appropriate permissions.

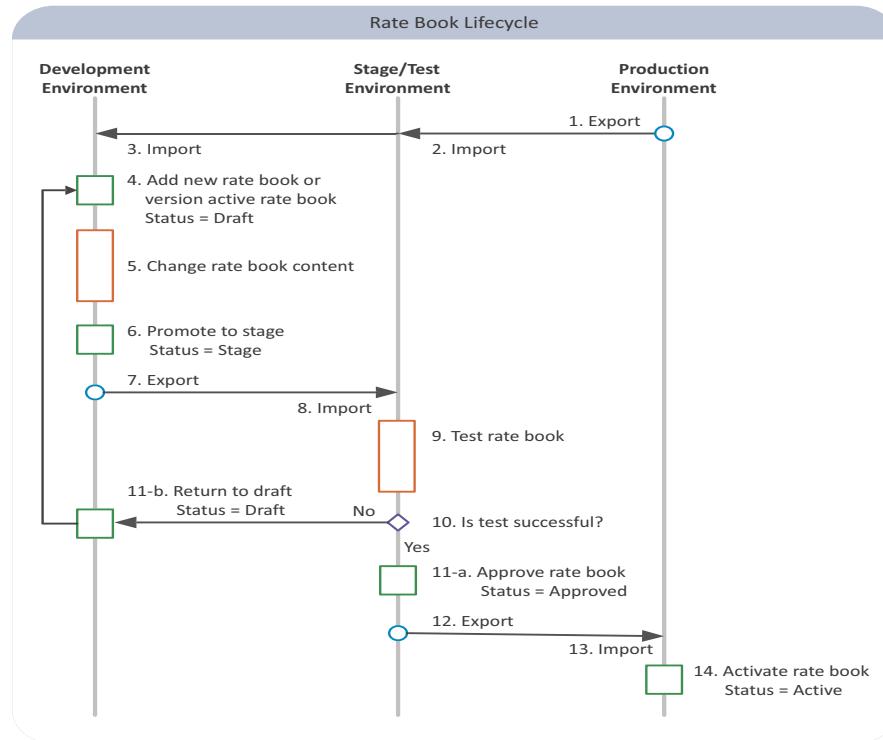
In the example, you can force Policy A to be re-rated with Rate Book v2. Simply initiate a policy change with the same effective date as the policy and change the rate-as-of date to later than 12/1/2010.

You can configure the policy rate-as-of date to meet other business requirements.

## Rate Book Lifecycle and Moving to Production

The default implementation includes a series of rate book statuses (**Draft**, **Stage**, **Approved**, and **Active**). The default implementation also includes the ability to easily move rate books from one environment to another to support typical rate book lifecycle requirements.

Most carriers have multiple environments (development, one or more stage/test environments and production) to better control production changes. The recommended lifecycle described in the following illustration assumes multiple environments. The lifecycle can also be applied to a single environment by skipping the export and import steps.



**IMPORTANT** If you modify rate books and their included rate tables and rate routines on multiple servers, Guidewire recommends that you set up unique public ID prefixes for those servers. These servers typically include development and stage servers. The unique public ID prefix ensures unique public IDs within objects of the same type. If the public ID prefixes are not unique, public ID clashes may occur when you import to the stage or production environment. For more information, see “Public ID Prefix” on page 82 in the *System Administration Guide*.

## Synchronize Development Rate Books with Production

Rate books, and their included rate tables and rate routines, are typically maintained and updated in a development environment then deployed to staging and finally to the production environment. Before making changes to the rate books in the development environments, make certain that you are modifying an exact copy of the rate books in the production environment.

For each rate book, the development environment must have all rate book editions, and all editions must be exact copies. If a rate table has not changed, newer rate books can refer to rate tables in older rate book editions.

To avoid conflicts when importing rate books to the production environment, you must do one of the following:

- If the rate books in the development environment are exact copies of the rate books in the production environment, no action is necessary. You can make rating changes in the development environment. Select this option if you are certain that the rate books in the development and production environments are the same.
- If the rate books in the development environment are not exactly the same as the rate books in the production environment, then export all relevant rate books from production. Next, import the rate books into your development environment. Then you can make rating changes in the development environment. Select this

option if there is a question whether the rate books in the development environment are a copy of the production rate books.

- Copy the production database into the development environment. Select this option when setting up an environment that needs production data. This process also ensures that the rate books in development and production environments match.

## Development Environment

In the development environment, developers edit rate books and the included rate tables and rate routines.

You can add a new rate book or create a new edition of an active rate book. These actions creates a new rate book in Draft status. The Draft status is the only status in which the rate book is open for edit. You can return to Draft status from either Stage or Approved if you need to make modifications. However, after a rate book is in Active status, you can no longer modify the rate book in any way, including a change in status.

The development environment can consist of one or more development servers. However, multiple servers increase the complexity and the potential for problems.

### One Development Server

If you have a single development server, one or more developers edit rate books, rate tables, and rate routines on the same server. The development server is the master copy from which you propagate changes to the stage and then production environments.

If you have a single development server, the public ID prefix of the development server need not be different than the public ID prefix of the staging server.

### Multiple Development Servers

If you have multiple development servers, developers edit rate books, rate tables, and rate routines on different servers. For example, one development server may be devoted to making changes to the rate books associated with a particular line of business.

To avoid overwriting the work of other development groups, devote a single development server to a discrete set of rate books and included rate tables and rate routines.

So that public IDs are unique within objects of the same type in the staging and production environments, each development servers must have a unique public ID prefix. Public ID collisions can occur if more than one development servers has the same public ID prefix. For more information about public Id prefixes, see “Public ID Prefix” on page 82 in the *System Administration Guide*.

## Stage Environment

After all changes are done, you promote the rate book from Draft to Stage status to indicate that data entry is complete. At this point the rate book is ready for testing. Testing can occur in the same development environment, but typically carriers have a separate staging environment for testing. If this separate environment exists, the rate book can be easily moved to that environment by using rate book export and import actions in the rate book editor. Exporting a rate book includes all included rate tables and definitions associated with that rate book.

**Note:** In a production environment, only Active rate books are considered for policy rating. However, a configuration parameter exists to allow testing of pre-Active rate books in a development or test environment. For more information, see “Minimum Rating Level Parameter” on page 495 in the *Configuration Guide*.

## Production Environment

After testing is complete, you can move the rate book to Approved status. You can then move the rate book to the production environment by using the same export/import actions. Then you can activate the rate book for

production use. Upon activation, the rate book status is set to Active and the rate book is available for production policy rating.

The default implementation includes sample user permissions to control who can move rate books to each stage. See “User Authority and Permissions for Rating Management” on page 497 in the *Configuration Guide* for details of the included roles and permissions. You can configure this to meet your specific needs.

## Rating Worksheets

For a quoted policy or policy transaction, the rating worksheet shows the actual values that the rate routine used to calculate the rate. For each rated object, the rating worksheet shows the following:

- Rate book code and edition
- Rate routine code and version
- Actual values used in the rate routine

Rating worksheets provide information for several different types of users or developers. You can use rating worksheets to debug rate routines. In submission policy transactions, underwriters can use rating worksheets to see how the policy is rated. The underwriter can decide whether to make an adjustment through a rating override or other mechanism. Auditors can use the information in rating worksheets to explain to a state regulator how a policy was rated. The auditor may need this information years after the policy rating occurred.

The following table shows the type of data a rating worksheet for a personal auto policy might contain:

	<b>Instruction</b>	<b>Result</b>	<b>Op</b>	<b>( Operand</b>	<b>) Operand value</b>
1	Liability - Bodily Injury and Property Damage				
2	pa_rtm_demo_rating (2)				
3	pa_cov_premium_rr (1)				
4	BaseRate	172.000	←	table:Base Rate	(PALiabilityCov, CA, 91007) = 172
5			R	.001	172.000
6	vehicleTypeFactor	10	←	table:Vehicle Type Factor	(CA, auto) = 10
7	vehicleCostFactor	0.44	←	table:Vehicle Cost Factor	(1000.00, CA) = 0.440000
8	AdjustedRate	258.825600	←	BaseRate	172.0000
9			x	table:CoverageFactor	(PALiabilityCov, PALiability, 250/500/100, CA) = 3.800000
10	...				

In row 1, the coverage being rated is Liability - Bodily Injury and Property Damage.

In row 2, The rate book code is pa\_rtm\_demo\_rating edition 2.

In row 3, the rate routine is pa\_cov\_premium\_rr version 1.

In rows 4 through 9, the Result and Operand Value columns show the actual values used by the rate routine.

### See also

- “Viewing Rating Worksheets” on page 577
- “Enabling Rating Worksheets in the Rate Routine Plugin” on page 518 in the *Configuration Guide*

## Impact Testing

You can use impact testing to see the impact that changing the rate book and rate routines have on policy premium for a group of policies. Impact testing generates test policy periods and rates them using the active rate books and selected comparison rate books. You can choose the test policy periods by product, jurisdiction, producer code, postal code, effective date, expiration date, and as of date.

In the default configuration, impact testing rates the baseline and test policy periods for all job types as submission jobs. You can configure impacting testing to rate a renewal as a renewal job rather than a submission job. For example, rate routines may include special steps for renewal jobs. Rate tables may contain factors based on job type.

In the default configuration, PolicyCenter prevents a renewal on a policy period which has an open renewal. If you make this configuration change, PolicyCenter can create a renewal on a policy period which has an open renewal. Making this configuration change impacts how PolicyCenter handles renewals in impact testing and in other areas of the product.

For information on configuring impact testing to rate renewals as renewal policy transactions, see “Rating Renewals as Renewal Jobs in Impact Testing” on page 516 in the *Configuration Guide*.

Impact testing creates two graphs. The X-axis is divided into impact ranges. Each range represents a percentage change to the policy premium, such as **No change** or **0% up to 5%**. In the **Policies Affected** graph, each bar represents the number of policies affected in each impact range. In the **Financial Impact** graph, each bar represents the monetary amount of change in each impact range.

The impact testing graphs only display changes to policy premium. The graphs exclude other costs, such as taxes, fees, and surcharges. However, you can configure impact testing to display other types of costs by modifying the `RatingExportUtil` Gosu class. In this class, change the `generateIStatistics` method.

You can export the coverage and cost comparisons to Excel. For each policy, the Excel spreadsheet shows details for each cost on the policy when rated using the active rate book and the comparison rate book. The details include:

- Baseline term amount
- Baseline actual amount
- Comparison term amount
- Comparison actual amount

### Impact Testing Warnings and Recommendations

Impact testing is designed to work in a test environment on a copy of production data. Impact testing is accessible only when the server is in development or test mode. Because impact testing affects system performance and creates test policy periods that persist in the database, impact testing is not accessible in production mode.

When you run impact testing, impact testing creates test policy periods in the database. In the test environment, observe the following warnings and recommendations:

- The work of users can interfere with impact testing results. During impact testing, Guidewire recommends that only the single user performing impact testing be logged into PolicyCenter.
- Disable integrations to other systems. If you run impact testing with integrations enabled, the integration can send test data to these production systems. For example, disable the free-text search integration that uses the search engine Solr. The same applies to other integrations such as an integration with a billing system. To avoid unnecessary costs, disable integrations to systems that charge access fees.
- While running impact testing batch processes, do not make changes in PolicyCenter that impact rating such as doing a policy change, canceling, or changing rate books. For example, the baseline creation batch process

or the test period quote batch process is running. During this time, quoting test periods fails if a user changes or cancels a policy which has a baseline.

- Do not include expired policies in the impact testing dataset. Specify an **In Force On** date to filter out expired policies.
- Impact testing excludes archived policies.
- Stop the Job Expire batch process to prevent PolicyCenter from unexpectedly expiring policies during or between impact testing runs. To manage batch processes on a test server, press ALT + SHIFT + T to display the **Server Tools** page, then select **Batch Process Info** from the **Server Tools** tab.

#### See also

- “Working with Impact Testing” on page 579
- “Impact Testing Plugin” on page 521 in the *Configuration Guide*
- “Work Queues for Impact Testing” on page 517 in the *Configuration Guide*

# Rating Management User Interface

This topic describes how to work with Guidewire Rating Management in the PolicyCenter user interface.

This topic includes:

- “Preparing to Use Rating Management” on page 551
- “Working with Rate Books” on page 551
- “Working with Rate Table Definitions” on page 557
- “Working with the Rate Table Editor” on page 562
- “Working with Rate Routines” on page 565
- “Working with Parameter Sets” on page 577
- “Working with Impact Testing” on page 579
- “Examples of Working with Rating Management” on page 583

## Preparing to Use Rating Management

You can access the Rating Management from the **Rating** submenu on the **Administration** tab of PolicyCenter.

Before you rate policies, you must enable the **PCRatingPlugin**. For instructions, see “Guidewire Rating Management and PCRatingPlugin” on page 411 in the *Integration Guide*.

In the base configuration, PolicyCenter has the permissions that enable you to work with Guidewire Rating Management. For more information, see “Assignment of Permission to Roles” on page 497 in the *Configuration Guide*.

## Working with Rate Books

You can search for, view, edit, copy, add, delete, import or export rate books.

- “Searching for a Rate Book” on page 552
- “Adding a New Rate Book” on page 552

- “Deleting a Rate Book” on page 552
- “Rate Book Details Screen” on page 553
- “Rate Book Status and Available Actions” on page 554
- “Rate Book Actions and Permissions” on page 555
- “Importing and Exporting Rate Books to XML” on page 556

## Searching for a Rate Book

1. Select **Administration** → **Rating** → **Rate Books**.

PolicyCenter displays the **Rate Books Search** screen.

2. Specify any of the following search parameters:

- **Policy Line** – <none selected> implies **any**
- **Code** – matches on rate book codes that contain the search criteria
- **Name** – matches on rate book names that contain the search criteria
- **Underwriting Company** – <none selected> implies **any**
- **Jurisdiction** – <none selected> implies **any**
- **Offering** – <none selected> implies **any**
- **Status** – <none selected> implies **any**
- **Effective Date** – matches on rate books with effective date greater or equal to the search criteria
- **Activation Date** – matches on rate books with activation date greater or equal to the search criteria

3. Click **Search**.

PolicyCenter displays the search results. The search results are ordered first by descending **Status** (**Active**, **Approved**, **Stage**, **Draft**), then by ascending **Policy Line** (alphabetically) and then chronologically by descending **Effective Date**.

4. Click the hypertext link in the **Code** column to view or edit any of the returned rate books in the **Rate Book Details** screen.

## Adding a New Rate Book

You can add new rate books by clicking **New Rate Book** in the **Search Results** pane or by viewing a rate book and clicking **Create New Edition**.

You must have the **ratebookview** and **ratebookedit** permissions.

PolicyCenter displays the **Rate Book Details** screen in edit mode. If you created this rate book by clicking **Create New Edition**, the rate book edition and the effective and expiration dates are blank. All other fields, including rate tables, are pre-populated with information from the rate book of which this rate book is a version. For more information about this screen, see “**Rate Book Details Screen**” on page 553.

## Deleting a Rate Book

You can delete a rate book if it is in **Draft** status.

1. Go to **Administration** → **Rating** → **Rate Books** and search for rate books.
2. If the rate book is in **Draft** status, select the check box in the left column of the rate book and click **Delete**. You are done.  
Otherwise, click the link in the **Code** column to view the existing rate book.
3. If the rate book is in **Stage** or **Approved** status, click **Return to Draft**. If the rate book is in **Active** status, you cannot return it to **Draft** status.

4. Click **Delete** to delete the rate book.

You must have the **ratebookview** and **ratebookedit** permissions.

## Rate Book Details Screen

Use the **Rate Book Details** screen to view and maintain rate books. You can only update rate books in **Draft** status. See “**Rate Book Lifecycle and Moving to Production**” on page 545 for more information about rate book statuses.

You must have the **ratebookview** and **ratebookedit** permissions.

The **Rate Book Details** screen is made up of several sections: the **Rate Book Details**, the **Included Rate Tables**, and the **Included Rate Routines**.

### Rate Book Details

The top of the **Rate Book Details** screen contains the following fields:

Field	Description
<b>Rate Book Details</b>	
<b>Code</b>	A code for the rate book. The combination of code and version must be unique within the specified policy line
<b>Name</b>	A human-readable name for the rate book. This field is localizable.
<b>Edition</b>	A string to represent the edition for the rate book. The format of the edition is specific to each carrier and may contain characters and/or numbers, such as 1.0, 2011, ISO Circular 4138:2010.
<b>Description</b>	A description for the rate book. This field is localizable.
<b>Status</b>	Set to <b>Draft</b> for new rate books. Set to other statuses based on rate book actions. See “ <b>Rate Book Status and Available Actions</b> ” on page 554.
<b>Last Status Change Date</b>	This field is set by the system and represents the date and time of the last status change. After activation of the rate book, this represents the activation date for the rate book. The field label changes depending on the status of the rate book. See “ <b>Selecting the Rate Book Edition During Policy Rating</b> ” on page 544 for more details on the use of activation date in the rating queries.
<b>Activation Date</b>	
<b>Last Updated By</b>	The user who last updated the rate book.
<b>Last Updated Time</b>	The time of the last update.
<b>Policy Criteria</b>	
<b>Underwriting Company</b>	A drop-down list of underwriting companies defined in PolicyCenter.
<b>Jurisdiction</b>	A drop-down list of jurisdictions defined in PolicyCenter.
<b>Policy Line</b>	A drop-down list of policy lines defined in the product model.
<b>Offering</b>	A drop-down list of offerings that depends on the policy line selected. The list includes all offerings defined in the product model for any product that includes the specified policy line.

---

<b>Policy Effective or Coverage Reference Date</b>	<p>The rate book's Policy Effective or Coverage Reference Date is compared to the reference date of the coverage. The reference date of the coverage can be set to:</p> <ul style="list-style-type: none"> <li>• The effective date of the policy term</li> <li>• The date that the coverable object was added to the policy within the current policy term</li> <li>• The date that the coverage was added to the policy within the current policy term</li> </ul> <p>In this field you specify the range that the reference date for the coverage effective date of the policy, coverable, or coverage must fall within:</p> <ul style="list-style-type: none"> <li>• <b>On or After</b> – The effective date of the policy must be equal to or greater than this date.</li> <li>• <b>Before</b> – The effective date of the policy must be less than this date.</li> </ul> <p>For more information about the reference date of the coverage, see "Specifying the Reference Date to Use" on page 91 in the <i>Product Model Guide</i>.</p>
<b>Renewal Effective Date</b>	<p>Specify the range that the policy renewal effective date must fall within:</p> <ul style="list-style-type: none"> <li>• <b>On or After</b> – The policy, coverable, or coverage reference date must be equal to or greater than this date.</li> <li>• <b>Before</b> – The effective date of the policy must be less than this date. You cannot set this date directly. PolicyCenter sets this value to the value of <b>Before in Effective Date</b>.</li> </ul>

---

## Included Rate Tables in a Rate Book

In the **Included Rate Tables** tab, you can specify which rate tables to include in this rate book.

On the left, the **Included Rate Tables** pane shows rate tables already included in the rate book. To remove an included rate book, select it and click **Remove from Rate Book**.

On the right, the **Available Rating Tables** pane shows rate table definitions available for inclusion in the rate book. This pane lists all rate table definitions defined for the policy line specified for the Rate Book and not already included. To include one, select it and click **Add to Rate Book**.

## Rate Routines Included in a Rate Book

In the **Included Rate Routines** tab, you can specify which rate routines to include in this rate book.

On the left, the **Included Rate Routines** pane shows rate routines already included in the rate book. The pane also displays the jurisdiction and version of the rate routine. To remove an included rate routine, select it and click **Remove from Rate Book**.

The **Available Rate Routines** pane shows rate routines available for inclusion in the rate book. This pane lists all rate routines defined for the policy line specified for the Rate Book and not already included. To include one, select the rate routine and version, then click **Add to Rate Book**. You can add only one rate routine with a particular code. For example, jurisdiction variants and versions of a rate routine all have the same code. You can add only one of these variants or versions.

## Rate Book Status and Available Actions

The default implementation includes the following rate book statuses.

---

Status	Description
Draft	<p>Initial status of new rate books. This status is the default status when you add or create a new version from another rate book. You cannot export a rate book that is in this status.</p> <p>Draft is the only status in which a rate book can be updated, including changes to included rate tables.</p> <p>Available actions: <b>Edit</b>, <b>Delete</b>, and <b>Promote to Stage</b> (if the rate book includes at least one rate table).</p>
Stage	<p>Data entry is complete and the rate book is ready for testing.</p> <p>Available actions: <b>Approve Rate Book</b>, <b>Return to Draft</b>, <b>Export to XML</b>.</p>

---

Status	Description
Approved	Testing is complete and the rate book is approved and ready to be moved to production. Available actions: <b>Return to Draft</b> , <b>Activate Rate Book</b> , <b>Export to XML</b> .
Active	The rate book is available for production use. Available actions: <b>Create New Edition</b> , <b>Export to XML</b> .

## Rate Book Actions and Permissions

Availability of rate book actions depends on the status of the rate book and user permissions.

The actions provided in the base configuration are:

Action	Description
View	View the Administration → Rating menu items. On the rate book <b>Search Results</b> , click the rate book code hypertext link to view the rate book. You must have the <b>ratebookview</b> permission.
New Rate Book	In Administration → Rating → Rate Books, click <b>New Rate Book</b> in Search Result to add a new rate book. You must have the <b>ratebookedit</b> permission.
Rate Book Details screen	
Edit	On the <b>Rate Book Details</b> screen, click <b>Edit</b> to edit the current rate book. The rate book must be in <b>Draft</b> status and you must have the <b>ratebookedit</b> permission.
Delete	On the <b>Rate Book Details</b> screen, click <b>Delete</b> to delete the current rate book. The rate book must be in <b>Draft</b> status and you must have the <b>ratebookedit</b> permission.
Promote to Stage	On the <b>Rate Book Details</b> screen, click <b>Promote to Stage</b> to change the status from <b>Draft</b> to <b>Stage</b> . The rate book must be in <b>Draft</b> status, must have at least one included rate table, and you must have the <b>ratebookedit</b> permission. Before changing the status, the system verifies that none of the included rate tables are empty.
Approve Rate Book	On the <b>Rate Book Details</b> screen, click <b>Approve Rate Book</b> to change the status from <b>Stage</b> to <b>Approved</b> . The rate book must be in <b>Stage</b> status and you must have the <b>ratebookapprove</b> permission.
Return to Draft	On the <b>Rate Book Details</b> screen, click <b>Return to Draft</b> to change the status back to <b>Draft</b> to make changes to the rate book or included rate tables. The rate book must be in <b>Stage</b> or <b>Approved</b> status and the user must have the <b>ratebookedit</b> permission.
Activate Rate Book	On the <b>Rate Book Details</b> screen, click <b>Activate Rate Book</b> to change the status from <b>Approved</b> to <b>Active</b> . The rate book must be in <b>Approved</b> status and user must have <b>ratebookapprove</b> permission. The system asks for a confirmation of this action before proceeding.
Create New Edition	On the <b>Rate Book Details</b> screen, click <b>Create New Edition</b> to create a new edition of the current rate book.  This action: <ul style="list-style-type: none"><li>• Creates a new rate book.</li><li>• Copies all of the rate book fields except version and effective and expiration dates.</li><li>• Creates referencing relationships to all included rate table versions.</li></ul> See “Managing Rate Books and Rate Tables” on page 543 for details on referencing relationships and versions. The rate book must be in <b>Active</b> status and you must have the <b>ratebookedit</b> permission.

**See also**

- “User Authority and Permissions for Rating Management” on page 497 in the *Configuration Guide*

## Importing and Exporting Rate Books to XML

You can export and import rate books to XML to move rate books from one environment to another without having to move the entire PolicyCenter database. The XML exported or imported includes the rate book and the rate routines associated with the rate book.

Validation checks that the imported rate book does not conflict with data already defined in the target environment.

### Precautions

Observe the following precautions to avoid public ID clashes or unexpected behavior:

- Create the initial development and test environments. Make certain that the new environment includes all relevant rate books. That is, include any book that contains a table that the new rate book will reference.
- To modify an active book, export the rate book to XML from the production environment to the stage/test environment. Have all users make edits in the stage environment. Guidewire recommends that developers not work on their rate book separately, then merge XML changes to a server. Developers can work on their Excel files separately, then upload those files to the stage environment.
- Make certain that each environment has a unique public ID prefix. See “Environment Parameters” on page 49 in the *Configuration Guide* for details.
- Do not change the exported XML before importing it into another environment. Do not edit the XML outside of PolicyCenter. Doing so may create unpredictable results.

### Export to XML

On Rate Book Details screen, click **Export to XML** to export the current rate book to XML before moving it to another environment.

The rate book can be in any status except **Draft**, and you must have **ratebookview** permission.

### Import from XML

On the **Search Results** pane of the Rate Book screen, click **Import from XML** to import an exported rate book.

The rate book cannot exist in PolicyCenter, and you must have the **ratebookview** and **ratebookedit** permissions.

The size of the imported file must less than or equal to the **MaximumFileUploadSize** parameter.

You can view import warnings or errors by clicking a link on the screen. These warnings and errors also appear in the log file.

**See also**

- “Deleting a Rate Book” on page 552
- “**MaximumFileUploadSize**” on page 48 in the *Configuration Guide*

### Import Validations

When importing from XML, PolicyCenter performs validations on the imported rate book.

#### Validations on the Imported Rate Book

PolicyCenter performs the following validation checks on the imported rate book:

- Activation date is not in the future
- The rate book does not conflict with an existing one, such as another rate book with the same code, version and policy line.
- The rate book's public ID is unique.
- The rate book contains at least one rate table and does not include any empty rate tables.
- The rate book status is not **Draft**.

#### Validations on Included Rate Tables

PolicyCenter performs the following validation checks on each included rate table:

- If a table references another table, validate that the owned table exists.
- The rate table definition for the imported rate table does not conflict with an existing rate table definition, such as having the same public id but different definition details.

#### Validations on Included Rate Routines

PolicyCenter performs the following validation checks on each included rate routine:

- Does the rate routine have a unique code and version?
- Does the rate routine include corrupted data, such as missing steps?
- All referenced functions exist.
- All data referenced through the parameter set exists.
- All rate table lookups have the required arguments.
- All typekey constants are valid.

PolicyCenter issues a warning if:

- The rate routine does not set required **CostData** fields.
- The type of data referenced by the parameter set changed.

## Working with Rate Table Definitions

You can search for, view, edit, copy, add, or delete rate table definitions.

- “Searching for a Rate Table Definition” on page 557
- “Adding a New Rate Table Definition” on page 558
- “Rate Table Definition Screen” on page 558

### Searching for a Rate Table Definition

**1. Select Administration → Rating → Rate Table Definitions.**

PolicyCenter displays the **Rate Table Definitions Search** screen.

**2. Specify any of the following search parameters:**

- **Policy Line** – Default is **any**.
- **Code** – Matches rate table codes that contain the search criteria.
- **Name** – Matches rate table names that contain the search criteria.

**3. Click **Search**.**

PolicyCenter displays the search results. The search results appear in alphabetical order, first by **Policy Line** and then by **Code**.

4. Click the hypertext link in the **Code** column to view or edit any of the returned rate table definitions in the **Rate Table Definition** editor.

## Adding a New Rate Table Definition

You can add new definitions by clicking **Add** in the **Search Results** pane or by viewing a similar definition and clicking **Copy**.

PolicyCenter displays the **Rate Table Definition Editor** in edit mode. If you used the **Copy** function, the rate table **Code** is blank and the word **Copy** appears at the beginning of the **Name**. All other fields are pre-populated with information from the copied definition.

## Rate Table Definition Screen

Use the **Rate Table Definition** screen to view and maintain definitions. This screen contains the following tabs:

- “Basics Tab of Rate Table Definition” on page 558
- “Parameters Tab of Rate Table Definition” on page 558
- “Factors Tab of Rate Table Definition” on page 560
- “Argument Sources Tab of Rate Table Definition” on page 561
- “Usage Tab of Rate Table Definition” on page 561
- “Rate Table Definition Validations” on page 562

### Basics Tab of Rate Table Definition

In the **Rate Table Definition** screen, the **Basics** tab has the following fields:

Field	Description
Code	A code for the definition. Must be unique within the specified policy line.
Name	A human-readable name for the definition. This field is localizable.
Description	A description for the definition. This field is localizable.
Policy Line	A drop-down list of policy lines defined in the product model.
Physical Table	The physical rate table/entity to map this definition to. Can use the generic physical table or reference a custom physical table. See “Physical Tables and Entities for Rate Table Definitions” on page 535 for details on generic and custom physical tables. To use the generic physical table, enter <b>DefaultRateFactorRow</b> .  In the base configuration, the <b>DefaultRateFactorRow</b> entity represents the default physical table. The base configuration also includes custom physical table represented by the <b>CoverageRateFactorRow</b> entity for parameters that are coverages.
Last Updated By	The name of the user who last updated this rate table. This field is read-only.
Last Update Time	The time that the user last updated this rate table. This field is read-only.

If you edit a rate table definition that is included in a rate book, you can update only the **Name** and **Description** fields.

### Parameters Tab of Rate Table Definition

In the **Rate Table Definition** screen, use the **Parameters** tab to maintain the parameter columns in the table definition. This tab contains a list of defined parameters at the top of the screen and a **Details** pane to display the details for the selected parameter in the list.

## Matching Rules

On the **Parameters** tab of a rate table definition, the **Matching Rule** for a parameter specifies how to compare a policy value to the parameter value in the rate table. In the default implementation, PolicyCenter includes the following matching rules:

- **Exact Match** – The policy value must match the parameter value exactly. For example, the coverage code or vehicle make must match exactly.
- **Range with Excluded Max** – The policy value must be greater than or equal to the minimum value in the range and less than the maximum value in the range. This rule is useful for ranges such as limits. For example, assume the ranges are 1 to 1,000,000 and 1,000,000 to 2,000,000. You do not have to specify 999,999.99 as a max value.
- **Range with Included Max** – The policy value must be greater than or equal to the minimum value in the range and less than or equal to the maximum value. Use this rule for ranges without any overlap, such as 16 to 25, 26 to 40, and 41 to 50.
- **Longest Substring Match** – Matches the parameter value which is the longest initial substring of the policy value.
- **Less Than Or Equal Match** – The policy value must be less than or equal to the parameter value. If multiple matches exist, the largest parameter value matches. For example, if a rate table has rows for values 1 through 10, and the policy value is 5.5, the row with parameter value 5 matches.
- **Greater Than Or Equal Match** – The policy value must be greater than or equal to the parameter value. If multiple matches exist, the smallest parameter value matches. For example, if a rate table has rows for values 1 through 10, and the policy value is 5.5, the row with parameter value 6 matches.
- **Interpolated Match** – This is the same as **Exact Match** if the policy value matches the parameter value exactly. If the policy value falls between two parameters values, then the interpolated factor is proportionally between the two factors.

For more information, see:

- “Rate Table Definition” on page 534, “Matching a Factor in the Rate Table” on page 536
- “Configuring Matching Rule Operations” on page 501 in the *Configuration Guide*

## Parameter Details Fields

The fields on the **Parameter Details** pane vary based on the parameter’s matching rule.

For **Exact Match**, **Greater Than Or Equal Match**, and **Less Than Or Equal Match** parameters, the screen displays one set of the parameter details fields.

For **Range Match with Included Max** or **Range Match with Excluded Max** parameters, the screen displays three sets of parameter fields. One defines the range (**Matching Rule**, **Code**, and **Data Type**). The second defines the minimum value (**Priority through Value Provider**). The third defines the max value (**Priority through Value Provider**).

The following table describes the fields on the **Parameters** tab.

Field	Description
Matching Rule	Set to the <b>Matching Rule</b> selected as part of the <b>Add</b> action. For a description of the matching rules, see “Matching Rules” on page 559.
Code	The name used to identify the parameter input value in rating queries. The name is required, and all parameter names must be unique within a rate table definition. PolicyCenter does not make certain that this name is unique across parameters and factors. However, Guidewire recommends that you define unique names across the whole rate table definition.
Data Type	A drop-down list of the data types in the physical table and supported by the <b>Matching Rule</b> . If the <b>Physical Table</b> is <b>DefaultRateFactorRow</b> , the <b>Exact Match</b> choices are: <b>String</b> , <b>Integer</b> , <b>Decimal</b> , <b>Boolean</b> , and <b>Date</b> . The <b>Range</b> choices are: <b>Integer</b> , <b>Decimal</b> , and <b>Date</b> .
Decimal Places	The number of decimal places that the user can enter when entering decimal parameters. This field applies to both values of a range parameter. This field only appears for the <b>Decimal</b> data type.

Field	Description
Priority	A numeric priority that controls the display order of the columns on this screen, in the Rate Table Editor and in exported Excel files.  The rate table displays the parameters from left to right in descending priority order. Lower numbers have higher priority, so that column 0 is higher priority than 10. Priority affects how factors in a rate table match. For more information, see "Matching a Factor in the Rate Table" on page 536.
Column Code	A descriptive name for the (logical) column.
Column Label	The label that appears as the column header in the Rate Table Editor and is exported to Excel files. This field is localizable so an entry field appears for each locale defined in the environment. A label for the default locale is required.
Display Type	A drop-down list to describe the width of the column for the Rate Table Editor. In the default implementation the choices are: Small, Normal, and Large.
Physical Column	The physical table column to map this logical column to. This is a drop-down list whose values depend upon the Physical Table defined on the Basics tab and the Data Type defined on this tab. The generic physical table has the following column choices: 8 string columns, 8 integer columns, 6 decimal columns, 2 date columns, and 2 boolean columns. The custom CoverageRateFactor table contains the following parameter columns: Coverage Code, Coverage Term Code, Coverage Term Option Code, and Jurisdiction.
Value Provider	Drop-down list of available value providers. Select one of the choices for Arguments, described in the following row.  You can add custom value providers. See "Configuring Value Providers" on page 499 in the Configuration Guide for more information.
Arguments	These are arguments to the value provider. For Typelist Value Provider, the Arguments are the list of type-lists defined in PolicyCenter.  For the other value provider types, the default implementation does not include any validation on this field, but you can configure it if required. The the custom value providers in the default implementation take the following arguments: <ul style="list-style-type: none"> <li>• Coverage Value Provider – Requires no arguments. Any argument values are ignored.</li> <li>• Coverage Term Value Provider – A single optional argument. <ul style="list-style-type: none"> <li>• Coverage Code for looking up coverage terms. If not specified, value provider implementation requires this parameter be dependent on another parameter. See Depends On in the following row.</li> </ul> </li> <li>• Coverage Term Option Value Provider - Two optional arguments. You must provide either none or both arguments. If no arguments are specified, the implementation requires this parameter be dependent on another parameter. See Depends On in the following row. <ul style="list-style-type: none"> <li>• Coverage Code – Code of the coverage in product model (configured for the policy line associated with the rate table).</li> <li>• Coverage Term Code – Code of the coverage term under the coverage.</li> </ul> </li> <li>• Termless Coverage Value Provider – Very similar to Coverage value provider but find only those coverages that have no terms.</li> <li>• Reference Factor Value Provider – Retrieves all distinct values used in a column of a rate table (source rate table). Needs two required arguments: <ul style="list-style-type: none"> <li>• Rate Table Code – Code of a source rate table.</li> <li>• Column Name – Name of the column in the source rate table.</li> </ul> </li> </ul>
Depends On	Defines a dependency from one column of a rate table to another. Existence of this dependency forces the target rate table column to perform post-on-change in the Rate Table Editor. For example, if one rate table column stores coverage terms, it has to do one of the following: <ul style="list-style-type: none"> <li>• Specify a predetermined value provider argument that represents the coverage. (See the description for the Value Provider.)</li> <li>• Depend on a column in the same rate table that stores coverage code. The drop-down list of choices contains the other parameters defined for the current rate table definition.</li> </ul>

## Factors Tab of Rate Table Definition

In the **Rate Table Definition** screen, use the **Factors** tab to maintain the factor columns in the table definition. A rate table can have a single factor or multiple factors.

In the default implementation, the look and functionality of the **Factors** tab is very similar to the **Parameters** tab with the following differences:

- The **Factors** tab includes the **String**, **Integer**, **Decimal**, **Boolean**, and **Date** data types.
- Factors do not have value providers.

**See also**

- “Rate Table with Multiple Factors” on page 538
- “Creating and Using a Rate Table with a Multiple Factors” on page 583

### Argument Sources Tab of Rate Table Definition

In the **Rate Table Definition** screen, use the **Argument Sources** tab to specify one or more argument source sets. An argument source set specifies a default argument source for each parameter in a parameter set. The default argument source is provided as a convenience for adding rate tables in a rate routine step. If you include a rate table in a rate routine step, and that rate table has a single argument source set, that source set automatically provides the default argument sources. If the rate table has more than one argument source set, you can select which source set to use. For each parameter, you can override the default argument source.

In the **Source of Argument Objects** field, specify the parameter set for this rate table definition. In the **Objects Available for Arguments** field, PolicyCenter displays the parameters in the selected parameter set.

In the **Source** panel, PolicyCenter displays the parameters which appear on the **Parameters** tab in the **Rate Table Definition** screen. For each parameter in the rate table, you can specify the **Argument Source**. The drop-down list for the **Argument Source** displays the entities accessible from the parameters that match the **Data Type**. The **Argument Source** list is also filtered by the value provider typelist specified on the **Parameters** tab.

In the **Source** panel, click **Add Source Set** to add another source set. If you have more than one source set, the first source set has **Name** set to **Default** and **Code** set to **DEFAULT**. Guidewire suggests that you change the name and code from **Default** and **DEFAULT** to more descriptive values. You can change these values as long as the rate table is neither in a promoted rate book nor in a rate routine.

For examples in the sample data, see the **BaseRate** and **DeductibleFactor** rate table definitions.

**See also**

- “Parameters Tab of Rate Table Definition” on page 558 for descriptions of the **Value Provider Type** and **Value Providers** fields

### Usage Tab of Rate Table Definition

In the **Rate Table Definition** screen, use the **Usage** tab to view the rate books that use this rate table definition. The page shows a tree view that lists in the first level all the rate books that own a particular version of the rate table. Then following each owning rate book at level two, you can see the list of rate books that reference that owned version of the rate table. See “Managing Rate Books and Rate Tables” on page 543 for more information.

On this tab, you can find out how many versions of the rate table exist and which rate books are affected by changes to this table. The view is in descending **Rate Book Status** order then descending **Rate Book Effective Date** order.

For the current rate table definition, the **Usage** tab has the following fields:

Field	Description
Rate Book Edition	The name and version of each rate book that uses this rate table definition. Click this link to go to the <b>Rate Book Details</b> screen for the selected rate book edition.
Rate Book Status	The status of the rate book.

Field	Description
Rate Book Effective Date	The effective date of the rate book.
Rate Table	<p>The relationship between the rate table version and the rate book.</p> <p>Values are:</p> <ul style="list-style-type: none"> <li>• Owned-Not Shared – This rate book is the first rate book to include this rate table. No other rate book includes this rate table.</li> <li>• Owned-Shared – Same as Owned-Shared except that another rate book includes this rate table.</li> <li>• Referencing – The rate book includes this rate table, but another rate book owns this rate table.</li> </ul> <p>These values are the same as in the <b>Usage</b> column in the Rate Book Details screen on the Included Rate Tables tab.</p> <p>Click the link in the Rate Table column to display the Rate Table screen for the rate book that uses it. Click the link in the Rate Book Edition column to display the Rate Book Details screen.</p>

### Rate Table Definition Validations

In addition to required field checks, the following requirements for rate table definitions are checked in the default implementation when you click **Update**:

- Rate table code is unique for the specified policy line.
- Specified Physical Table is valid – If a custom physical table is defined instead of the default table, PolicyCenter ensures that the custom table is defined correctly:
  - Must have a foreign key attribute pointing to the RateTable entity.
  - The foreign key must be named RateTable.
- At least one parameter has been defined.
- At least one factor has been defined.
- Parameter names are unique within the rate table definition.
- Parameter column names are unique within the rate table definition.
- Parameter priorities are unique within the rate table definition.

## Working with the Rate Table Editor

Use the **Rate Table** editor to view and edit the content of rate tables. You always edit rate tables in the context of a rate book. This requirement assures that the system can correctly manage the versions of rate tables and their relationships to rate books. See “Managing Rate Books and Rate Tables” on page 543 for more information.

- “Selecting a Rate Table” on page 562
- “Rate Table Screen” on page 563
- “Editing Rate Table Content in PolicyCenter” on page 563
- “Editing Rate Table Content in Excel” on page 564
- “Rate Table Update Validations” on page 565
- “Excel Rate Table Import Validations” on page 565

### Selecting a Rate Table

There are two ways to select a rate table for viewing or editing:

- From the **Rate Book Details** screen
- From the **Rate Table Definition → Usage** tab

**To select a rate table from the Rate Book Details screen**

1. Select a rate book to view from the Rate Book Search Results pane to go to the Rate Book Details screen.
2. In the Included Rate Tables list, click the hypertext link in the Name column to view or edit the selected rate table in the Rate Table editor.

**To select a rate table from the Rate Table Definition Editor Usage tab**

1. Select a rate table definition from the Rate Table Definition Search Results pane to go to the Rate Table Definition editor.
2. Go to the Usage tab.
3. Click the hypertext link in the Rate Table column to go to the Rate Table editor for the selected rate table version.

## Rate Table Screen

Use the Rate Table screen to view the content of rate tables. You can edit the rows if the rate book that owns this version is in Draft status.

The rate table contains the following read-only fields to show the key information for the rate table definition and rate book related to the rate table:

Field	Description
Code	The rate table definition code.
Name	The rate table definition name in the default locale.
Description	The rate table definition description in the default locale.
Policy Line	The rate table definition policy line.
In Rate Book	The rate book name in the default locale.
Usage	Displays Referencing, Owned-Shared, or Owned-Not Shared to describe the relationship between the rate table version and the rate book. Values are: <ul style="list-style-type: none"><li>• Owned-Not Shared – This rate book is the first rate book to include this rate table. No other rate book includes this rate table.</li><li>• Owned-Shared – Same as Owned-Shared except that another rate book includes this rate table.</li><li>• Referencing – The rate book includes this rate table, but another rate book owns this rate table.</li></ul>

On the Rate Table screen, you can view rows for the rate table in the Rate Table Content section.

PolicyCenter displays the fields on this screen based on the related rate table definition parameter and factor details such as column label, priority, and display type. When adding or updating rows, the value provider details for each parameter control the data entry field for that parameter. See “Parameters Tab of Rate Table Definition” on page 558 for more information.

If the rate book that owns this version is in Draft status, then you can edit the rows. For more information, see “Editing Rate Table Content in PolicyCenter” on page 563.

You can also maintain the rate table content in Microsoft Excel. For more information, see “Editing Rate Table Content in Excel” on page 564.

## Editing Rate Table Content in PolicyCenter

You can edit a rate table if the rate book is in Draft status.

1. In PolicyCenter, navigate to Administration → Rating → Rate Books.

2. Click **Search** and select a rate book in **Draft** status in the **Search Results**. PolicyCenter displays the **Rate Book Details**.
3. Click to display one of the **Included Rate Tables**. PolicyCenter displays the **Rate Table** screen.
4. To edit rate table content in PolicyCenter, click **Edit**. PolicyCenter displays the page in edit mode. If there is no **Edit** button, then the rate book is not in **Draft** status.
5. On this screen, you can:
  - Add new rows by clicking **Add**. A blank row appears at the bottom of the current page. After you complete your entry, click **Update** to save the added row.
  - Update existing rows by changing any of the fields and clicking **Update**.
  - Remove existing rows by selecting one or more rows and clicking **Remove**.

See “Rate Table Update Validations” on page 565 for details on the validation that is done on update.

## Editing Rate Table Content in Excel

1. In PolicyCenter, navigate to **Administration** → **Rating** → **Rate Books**.
2. Click **Search** and select a rate book in the **Search Results**. PolicyCenter displays the **Rate Book Details**.
3. Click to display one of the **Included Rate Tables**. PolicyCenter displays the **Rate Table** screen.
4. To edit rate table content in Microsoft Excel, click **Export to Spreadsheet** to create a structured Excel file containing the current contents of the table. The Excel file is in **.xslx** format. The file name is the same as the rate table code.

This action creates the structured Excel file and ask you to **Open** or **Save** the file. You can either:

- Open and begin editing, then save the file after you are done.
- Save, then open, edit, and save again later.

The structure of the exported Excel file enables import back into PolicyCenter. To avoid import errors, do not modify the structure, headers or order of the exported columns. You can add new rows or update existing rows. You can add new columns after the last exported column. These columns are ignored on import. For your convenience, the exported file includes some columns such as display names for codes. These columns appear in grey and are ignored on import.

In Excel, do not insert a hyperlink into a field. If you do this, PolicyCenter throws an **IllegalStateException** when you import the Excel file.

**Note:** The following instructions show you how to import the Excel file back into PolicyCenter. Importing from Excel requires that the rate book be in **Draft** status.

5. After you complete and save your edits in Excel, return to PolicyCenter and display the rate table you want to import into in the **Rate Table** screen.
6. From the **Rate Table** screen, click **Import From Excel**. Specify the file to import. The file must be in **.xlsx** format. You can either type the file name directly or use the **Browse** function to search for and select the file that you saved previously.  
PolicyCenter imports and displays the contents of the imported Excel file in edit mode. You can fix any errors and visually verify the imported data. At this point, the data has not been saved in PolicyCenter. See “Excel Rate Table Import Validations” on page 565 for details on the validation that is done on import.
7. Click **Update** to save. PolicyCenter performs the same update validation on the imported data as when edits are made directly in the PolicyCenter application. See “Rate Table Update Validations” on page 565 for details of that validation.

## Rate Table Update Validations

When you click **Update** after modifying rate table content, PolicyCenter performs the following validations:

Validation	Description
Min/Max	If two parameters make up a range operation, check that the value for the min column is: <ul style="list-style-type: none"><li>• For <b>Range Max Excluded</b> – Less than the value of the max column.</li><li>• For <b>Range Max Included</b> – Less than or equal to the value of the max column.</li></ul>
Range overlap	Do not allow a range in one row to be a subset or a disjointed set from a range in a different row. For example, row 1: 2 - 5, row 2: 4 - 8.
Duplicate row	Do not allow a row to contain the same parameter values as another row.

### See also

- “Editing Rate Table Content in PolicyCenter” on page 563
- “Editing Rate Table Content in Excel” on page 564

## Excel Rate Table Import Validations

PolicyCenter performs the following validations on **Import** from Microsoft Excel:

- Is the file based on an Excel workbook that PolicyCenter exported? The validation checks that the headers are correct and in the right order.
- Does the file have the same column count for the header row as the rate table? The validation ignores display-only fields.

A violation causes the import to fail immediately.

PolicyCenter also ensures that data values for parameters with value providers adhere to the value provider edit rules. Any violations of these rules does not cause the import to fail, but the invalid value is blanked out and an error message appears on the **Rate Table Editor** page.

### See also

- “Editing Rate Table Content in PolicyCenter” on page 563
- “Editing Rate Table Content in Excel” on page 564

## Working with Rate Routines

In PolicyCenter, you can create and edit rate routines.

- “Accessing Rate Routines” on page 566
- “Adding a New Rate Routine” on page 566
- “Deleting a Rate Routine” on page 567
- “Actions on Rate Routines” on page 567
- “Adding Steps to a Rate Routine” on page 568
- “Instruction and Operand Types” on page 570
- “Specifying a Function as the Operand in a Rate Routine Step” on page 576
- “Creating a Rate Routine for Another Jurisdiction” on page 576
- “Specifying a Flat-rated Coverage in a Rate Routine” on page 576
- “Viewing Rating Worksheets” on page 577

## Accessing Rate Routines

You must have the `ratebookview` permission to access rate routines.

1. In PolicyCenter, select **Administration** → **Rating** → **Rate Routines**.

This screen has **Code** and **Name** fields.

The search returns all routines that contain the search criteria. The search is not case-sensitive. For example, if you enter `premium` in **Name**, the search returns all routines that contain that string, such as the **PA Coverage Premium Algorithm** routine.

To view all rate routines, leave **Code** and **Name** blank.

2. Enter your search criteria, then click **Search**.

For each search result, PolicyCenter displays the **Name**, **Code**, and **Version**.

## Adding a New Rate Routine

You must have the `ratebookview` and `ratebookedit` permissions to add a new rate routine. After adding the rate routine in PolicyCenter, add the rate routine to a rate book and to the rating plugin as described in step 6 and step 7.

1. To add a rate routine, you can add a new one or create a copy from an existing one. In either case, the rate routine version is 1. PolicyCenter creates a new `CalcRoutineDefinition` entity instance that represents the rate routine.

- To add a new rate routine, go to the **Rate Routines** screen, and click **Add** under **Search Results**. PolicyCenter creates a new rate routine with no steps.
- To copy a rate routine, go to an existing rate routine, and click **Copy**. PolicyCenter creates a copy of the rate routine.

2. Enter the following information for the rate routine:

Field	Description
<b>Code</b>	A unique code for the routine. This is the <b>Code</b> on the <code>CalcRoutineDefinition</code> entity instance.
<b>Name</b>	A name for the routine. This is the <b>Name</b> on the <code>CalcRoutineDefinition</code> entity instance.
<b>Jurisdiction</b>	The jurisdiction to which this routine applies. If not specified, this routine applies to jurisdictions that do not have a jurisdiction variant.
<b>Version</b>	Read-only. A version for the routine. This is the <b>Version</b> on the <code>CalcRoutineDefinition</code> entity instance.
<b>Description</b>	A description for the routine. This is the <b>Description</b> on the <code>CalcRoutineDefinition</code> entity instance.
<b>Line of Business</b>	The line of business that this routine applies to.
<b>Parameter Set</b>	Select a parameter set. The drop-down list displays the parameter sets defined for the selected <b>Line of Business</b> .
<b>Parameters in Parameter Set</b>	Read-only. The parameters in the selected <b>Parameter Set</b> . If you hover the mouse over the parameters, PolicyCenter displays the value in the <b>Type</b> column for each parameter. For more information, see “Adding, Changing, or Removing Parameters from a Parameter Set” on page 578.
<b>Last Updated By</b>	The name of the user who last updated this rate routine. This field is read-only.
<b>Last Update Time</b>	The time that the user last updated this rate routine. This field is read-only.

3. Add or update steps as described in “Adding Steps to a Rate Routine” on page 568.

4. If you want to save your work but continue editing, click **Apply**. The rate routine remains in edit mode, and you can continue making changes to it.

PolicyCenter displays a validation message if the **Code** and **Version** are not unique.

**IMPORTANT** Validate all rate routines before promoting a rate book. To validate a rate routine, click the **Validate** button in the **Steps** panel of the rate routine screen while in edit mode. PolicyCenter displays validation errors and warnings as appropriate. Errors prevent rate book promotion to Stage status. Warnings, however, do not prevent promotion. This behavior is intentional because validation warnings do not necessarily indicate a problem with the rate routine. Test rate routines thoroughly – some warnings can cause run-time errors when rating a policy transaction.

5. After you finish editing, click **Update** to save your work. The rate routine is no longer in edit mode. If PolicyCenter finds validation issues, the user receives a validation message and does not save the rate routine changes.
6. Add the rate routine to a rate book. See “Rate Routines Included in a Rate Book” on page 554.
7. Add the rate routine to the rating engine for the line of business. See “Configuring the Rating Engine to Execute the Rate Routine” on page 503 in the *Configuration Guide*.

## Deleting a Rate Routine

You must have the `ratebookview` and `ratebookedit` permissions to delete a rate routine.

1. To delete a rate routine, navigate to **Administration** → **Rating** → **Rate Routines**.

Rate routines that you can delete have a checkbox in the first column of the **Search Results**. You cannot delete a rate routine if it is in a rate book with a **Status** of **Active**, **Approved**, or **Stage** includes it.

2. To delete a rate routine version, select the checkbox next to the rate routine and click **Delete**.

## Actions on Rate Routines

You must have the `ratebookview` and `ratebookedit` permissions to perform these actions.

In PolicyCenter, you can take the following actions on rate routines on the **Edit Rate Routine** screen:

- **Edit** – Edit the current rate routine. Unlike rate tables, PolicyCenter does not validate rate routines on **Update**. To validate a rate routine, click **Validate**. If the rate routine is in an active rate book, you can only change the name and description.

**IMPORTANT** Validate all rate routines before promoting a rate book. PolicyCenter displays validation errors and warnings as appropriate. Errors prevent rate book promotion to Stage status. Warnings, however, do not prevent promotion. This behavior is intentional because validation warnings do not necessarily indicate a problem with the rate routine. Test rate routines thoroughly – some warnings can cause run-time errors when rating a policy transaction.

- **New Version** – Increment the version and open it for editing. For more information, see “Rate Routine Versions” on page 541.
- **Create Jurisdiction Variant** – Create a copy of the current rate routine for a particular jurisdiction. The copy is set to version 1 and has the same **Code** which you cannot change. Set the **Jurisdiction** to the new jurisdiction. Make other changes to the new rate routine. For more information, see “Rate Routine Variant Identifiers” on page 541.
- **Copy** – Copy the current rate routine version to a new rate routine. You must specify a new rate routine code.

Rate routines are imported or exported to XML when you import or export a rate book.

### See also

- “Importing and Exporting Rate Books to XML” on page 556

## Adding Steps to a Rate Routine

You can add steps to a new rate routine. You can add steps to an existing rate routine if it is not in use by a promoted rate book. The **Steps** panel of the rate routine screen has the following buttons:

Field	Description
Append	Select <b>Add 1 Row</b> or <b>Add 10 Rows</b> from the drop-down list to add rows to the end of the steps.
Insert	This drop-down list is active if you select one or more rows. Select from the following choices on the list: <ul style="list-style-type: none"> <li>• <b>Duplicate</b> – Inserts a copy of the selected step immediately after the selected step. If you select multiple steps, inserts a copy of the selected steps immediately after the last step in the selection.</li> <li>• <b>Insert Before</b> – Inserts a blank step before each selected step.</li> <li>• <b>Insert After</b> – Inserts a blank step after each selected step.</li> </ul>
Remove Row	This button is active if you select one or more rows. Removes the selected rows.
↑	This button is active if you select one or more rows. Moves each selected row up one position.
↓	This button is active if you select one or more rows. Moves each selected row down one position.
Validate	This button is active in edit mode. Validates each step and the rate routine as a whole.

### Fields in a Rate Routine Step

Each rate routine has one or more instructions that define the rating algorithm. Each instruction is composed of one or more steps. This instruction is not the same as the **Instruction** field described in the following table.

Each step in the rate routine appears as a row in the **Steps** panel. Each step has the following fields:

Field	Description
Checkbox	Select the current step. Appears in edit mode only.
#	The step number. Does not appear in edit mode.
Instruction	Optionally, select a value. The instruction field is the target of an assignment operator, a conditional, or a section comment. Select <b>Clear</b> to remove the value of this field.  If the instruction field is the target of an assignment operator, you can set the <b>Instruction</b> field to one of the following: <ul style="list-style-type: none"> <li>• Properties on the cost</li> <li>• Parameters, including rate modifiers</li> <li>• Variables</li> </ul> For more information, see “Instruction and Operand Types” on page 570. Some rate routines do not calculate properties on the cost, so these properties do not appear as choices. For more information, see “Rate Routines That Do Not Calculate Properties on the Cost” on page 541.
Op	The operator in a rate routine step can be one of the following: <ul style="list-style-type: none"> <li>• Blank</li> <li>• Assignment – Assign the operand to the <b>Instruction</b> field.</li> <li>• Arithmetic operators – Includes multiply, divide, add, and subtract.</li> <li>• Rounding operators – A rounding operator rounds the results of the previous expression to the specified scale. A step with a rounding operator must be last step of an expression.</li> </ul> For more information, see “Operators in Rate Routine Steps” on page 569.
(	Add one or more opening parentheses to group a series of steps. The closing parenthesis may occur in a later step.

Field	Description
Operand	<p>Select one of the following operand types:</p> <ul style="list-style-type: none"> <li>• Properties on the cost</li> <li>• Functions</li> <li>• Parameters</li> <li>• Rate tables</li> <li>• Variables</li> <li>• Conditional expressions</li> <li>• Constants</li> <li>• Date constants</li> <li>• Typelist values</li> <li>• Scale</li> </ul> <p>For more information, see “Instruction and Operand Types” on page 570.</p> <p>Some rate routines do not calculate properties on the cost, so these properties do not appear as choices. For more information, see “Rate Routines That Do Not Calculate Properties on the Cost” on page 541.</p>
)	Add one or more closing parentheses to group a series of steps. The opening parenthesis may occur in a previous step.
Line Comment	Enter a brief description of the step.

A step can contain a section comment or be blank. Use a section comment to provide comments about the following or preceding rows. Use a blank step to improve readability. You can add a **Line Comment** to a blank step.

## Operators in Rate Routine Steps

The operator in a rate routine step can be an assignment, arithmetic, or rounding operator or blank.

### Blank Operator

The operator can be blank if it is unspecified or if the **Instruction** field contains a conditional.

### Assignment Operator

There is one assignment operator:

- $\leftarrow$  – Assign the operand to the instruction.

### Arithmetic Operators

The arithmetic operators are:

- $\times$  – Multiply the previous expression with the operand.
- $\div$  – Divide the previous expression by the operand.
- $+$  – Add operand to the previous expression.
- $-$  – Subtract the operand from the previous expression.

### Rounding Operators

A rounding operator rounds the results of the previous expression to the specified scale. A step with a rounding operator must be last step of an expression. In the base configuration, the rounding operators are:

- **R** – Round half up. Round towards the nearest number according to scale. If both numbers are equidistant, round up. Always round away from 0.
- **RD** – Round down. Round down to the nearest number according to scale. Always round towards 0.
- **RU** – Round up. Round up to the nearest number according to scale. Always round away from 0.
- **RE** – Round half-even. Round towards the nearest number according to scale. If both numbers are equidistant, round towards the even number.

The following table contains some examples of rounding.

Operator	Scale	Value	Rounded value	Description
R	1	10.3	10	Round 10.3 half up with a scale of ones. The number 10.3 is closer to 10 than 11, therefore round to 10.
R	1	10.5	11	The number 10.5 halfway between 10 and 11, therefore round to 11.
R	1	-10.3	-10	The number -10.3 is closer to -10 than -11.
R	1	-10.5	-11	The number 10.5 is halfway between -10 and -11, therefore round away from 0 to -11.
RD	1	10.9	10	Round down 10.9 towards 0. Therefore, round down 10.9 to 10.
RD	1	-10.9	-10	Round down negative numbers towards 0. Therefore, round down -10.9 to -10.
RU	.01	5.551	5.56	Round up away from 0. Therefore, round up 5.551 to 5.56.
RU	.01	-5.551	-5.56	Round up away from 0. Therefore, round up -5.551 to -5.56.
RE	.01	5.551	5.55	Round half-even towards the nearest number.
RE	.01	5.545	5.54	Round half-even towards the even number since 5.545 is halfway between 5.54 and 5.55.
RE	.01	-5.555	-5.56	Round half-even towards the even number since -5.555 is halfway between -5.55 and -5.56.

Through configuration, you can also add rounding types for ceiling, floor, half down, and unnecessary.

#### See also

- Scale in “Instruction and Operand Types” on page 570
- “Fields in a Rate Routine Step” on page 568
- “Configuring Rounding Operators” on page 505 in the *Configuration Guide*

## Instruction and Operand Types

You can specify a number of types for the instruction and operand fields. To distinguish the rate table type, rate tables appear with a **table:** prefix in the base configuration. The prefix **table:** precedes the name of the rate table in a rate routine step. The rate table prefix distinguishes rate tables from variables. Without the prefix, rate tables with no **New Argument Source / Value** appear similar to variables.

The following table shows which types are available for the instruction and operand fields.

Instruction types	Operand types	Prefix	Return value
Properties on the Cost	Properties on the Cost	Not specified	Any type
Parameters	Parameters		Any type
Variables	Variables	Not specified	Any type
Conditional Instructions			Not applicable
	Conditional Expressions		Boolean
	Functions	Not specified	Any type
	Rate Tables	table:	String, integer, decimal, Boolean, or date
	Constants		Numeric, Boolean, string, or null
	Date Constants		Date

Instruction types	Operand types	Prefix	Return value
	TypeList Values		Typekey
	Scale		Numeric
Section Comment			Not applicable

Some rate routines do not calculate properties on the cost, so these properties do not appear as choices. For more information, see “Rate Routines That Do Not Calculate Properties on the Cost” on page 541.

In the base configuration, only rate tables have a prefix specified. For properties on the cost, variables, functions, and rate tables, you can add or change the prefix that appears in the step. For more information, see “Configuring Prefixes that Identify Types in Rate Routine Steps” on page 505 in the *Configuration Guide*.

### Drop-Down Lists for the Instruction and Operand Fields

The **Instruction** drop-down list contains properties or local variables that are writable.

The items in the **Operand** drop-down list depend upon on the **Instruction** and **Op** selections:

- If the **Instruction** is the assignment operator, PolicyCenter displays operands of the same type as the **Instruction**.
- If the step has an arithmetic operator, PolicyCenter displays operands of a type compatible with the arithmetic expression.
- If the step has a rounding operator, PolicyCenter displays scale operands.
- If you do not specify **Instruction** and **Op**, then PolicyCenter displays operands of all types. The **Constant** operand is checked but grayed out indicating that you can type a constant directly into the **Operand** text field.

For example, you have a step with an **Instruction** set to **Base Rate** which is a numeric value. In the **Operand** drop-down list, **Conditional** is grayed out because **Conditional** returns a Boolean. The **Operand** drop-down list contains **Parameters**, but only lets you select numeric types.

### Properties on the Cost

You can choose properties on the cost in both **Instruction** and **Operand** fields. These properties are often numeric but can be of any type. For example, **ProrationMethod** is a typelist field that is a property on cost.

Some rate routines do not calculate properties on the cost, so those properties do not appear as choices. For more information, see “Rate Routines That Do Not Calculate Properties on the Cost” on page 541.

In an **Instruction** or **Operand** field, you can choose these properties on the cost:

- **AdjustedRate** – The adjusted rate.
- **BaseRate** – The base rate.
- **Basis** – The basis.
- **ProrationMethod** – Set the proration method for the rate routine to **Day-based pro rata amount** or **Flat**. The default value is **Day-based pro rata amount**. You can set the proration method in any step. Guidewire recommends that a rate routine only set one proration method for a given coverage.

For more information about flat-rated coverages, see “Prorated or Flat Costs” on page 421 and “Specifying a Flat-rated Coverage in a Rate Routine” on page 576.

- **TermAmount** – The term amount.

The properties on the cost correspond to properties on a **Cost** delegate such as **BACost** for business auto costs. All **Cost** objects have a **Basis** and **ProrationMethod** property. The **AdjustedRate**, **BaseRate**, and **TermAmount** correspond to properties on the **Cost** object. For example, the **Cost** object has **ActualAdjRate**, **StandardAdjRate**, and **OverrideAdjRate** properties. The **BaseRate** and **TermAmount** have similarly named **Actual**, **Standard**, and **Override** properties.

For a line that does not support overrides the **AdjustedRate**, **BaseRate**, and **TermAmount** correspond to the Standard amounts. In the base configuration, personal auto does not support overrides.

For a line that supports overrides, the behavior varies depending on:

- How the rating engine handles a particular type of cost
- Whether there is an override on that cost
- Whether it is doing one-pass or two-pass handling of overrides

In the base configuration, commercial property supports overrides.

For more information, see “Rating Overrides” on page 445 and “What Do Cost Data Objects Contain?” on page 422 in the *Integration Guide*.

## Parameters

You can choose Parameters in both **Instruction** and **Operand** fields.

The parameters in the parameter set for the rate routine determine the parameters that you can access. The parameters provide access to objects on the policy, including rate modifiers on the policy line and coverages.

For both the **Instruction** and **Operand** fields, the current step filters the parameters.

For the **Instruction** field, you can choose parameters in the parameter set that are writable. If a parameter is writable, you can select its writable properties. In the **Instruction** field, you can select writable properties directly attached to the parameter. You cannot select properties on subobjects of the parameter. In the sample data, the **PADriverAssignmentParamSet** has a **DriverAssignmentInfo** parameter which is a writable.

For the **Operand** field, the types of parameters subobjects and properties that you can access are:

- Any type of property
- Subobjects not accessed through an array or list

To access a subobject that does not meet this criteria, you can create a function or enhancement property.

### Filtering Parameter Set Fields of the Same Type When Entering Parameters in a Rate Routine

When selecting parameters in a rate routine step, PolicyCenter filters subfields of a field that is the same type as another parameter in the parameter set.

For example, the **PA Coverage Parameter Set** parameter set in the sample data includes these two parameters:

- **PolicyLine** with type `entity.PersonalAutoLine` maps to a **PersonalAutoLine** object
- **Vehicle** with type `entity.PersonalVehicle` maps to a **PersonalVehicle** object

In a rate routine that uses this parameter set, you cannot access properties on a **PersonalVehicle** object through the **PolicyLine** parameter in a rate routine step. You can access the **PersonalVehicle** objects on the **PolicyLine**, but you cannot access the properties that the **PersonalVehicle** contains. If you require access to a property on the object, then you can explicitly define that as a parameter. For example, define a parameter that maps to `PersonalVehicle.LicensePlate`.

Follow these steps to see this filtering in the **PA Coverage Premium Algorithm** rate routine.

1. Go to **Administration** → **Rate Routines** and select **PA Coverage Premium Algorithm**.

2. Click **Create New Version**.

3. In **Steps**, select **Append** → **Add 1 Row**.

PolicyCenter adds a row at the end of the rate routine.

4. Go to the new row.

5. In the **Operand** field, select **Parameters** → **PolicyLine**.

PolicyCenter displays the **Select a Policy Data Field** screen.

6. Click **Next** until you arrive at the page displaying **Vehicle1** and **Vehicle2**, both of type **PersonalVehicle**.

Notice that PolicyCenter displays these two fields, but not subfields such as **Vehicle1.AnnualMileage**. You can access **AnnualMileage** through the **Vehicle** parameter.

### Accessing Rate Modifiers in Parameters

If the parameter set for a rate routine includes a parameter for the policy line, then you can access rate modifiers on the policy line. The rate modifier is usually multiplied with the prior expression.

In some cases, the rate modifier is zero-based, with zero indicating no modification to the rate. In the rate routine, add one to the value to avoid multiplying by zero and setting the rate to zero:

```
1 + rate_modifier
```

In other cases, the rate modifier value is calculated prior to rating and available on the policy line.

In the sample data, the **CP Building Coverage Premium Algorithm** rate routine contains examples of these two types of rate modifiers. For example, the rate routine adds the 1 to the **CPScheduleCredits** modifier. The rate routine does not, however, add 1 to the **ProductModifierFactor** modifier. Instead, the **ProductModifierFactor** modifier value is calculated in **CommercialPropertyLineEnhancement.gsx** in the **gw.rating** package.

**Note:** In the base configuration, the parameter sets in the small and large sample data sets include a policy line parameter, **Policy Line**. For more information, see “Adding Policy Line Rate Modifiers to a Parameter Set” on page 579.

### Accessing Coverages in Parameters

You can access coverage terms, options, and packages in parameters if the parameter set for the rate routine contains a parameter where both of the following are true:

- The **Type** field specifies a coverage
- The **Coverage** field specifies a coverage pattern

When accessing a coverage term, you can select its **code**, **covterm**, or **value**. (The **Instruction** type filters the selection. To see **code**, **covterm**, and **value**, set the **Instruction** to a new variable.) The **code** is the coverage term code. The **covterm** is the coverage term object. The **value** is the value of the coverage term.

In the sample data, the **PIP NJ Basic Algorithm** rate routine accesses coverage terms on the **PIPNJCovCoverage** parameter. This rate routine uses the parameter set named **PIP NJ Parameter Set**.

## Variables

Variables are available in both **Instruction** and **Operand** fields. Variables are implicitly typed. Therefore, you can assign any value to the variable, and the **Operand** drop-down list always displays variables.

In an **Instruction** field, you can define a variable. In an **Instruction** or **Operand** field, you can select a previously defined variable. The scope of these variables is the current rate routine.

When you add a variable, the name must be a valid Gosu variable name and cannot be the same as an in-scope object, such as **BaseRate**.

## Conditional Instructions

You can specify a conditional in the **Instruction** field only.

The conditionals are:

- **IF** – Begin a conditional instruction. An **ENDIF** must follow this conditional.
- **ELSEIF** – Continue a conditional instruction. An **IF** must precede this conditional.
- **ELSE** – Continue a conditional instruction. An **IF** must precede this conditional. Can follow **ELSEIF**.

- **ENDIF** – Ends a conditional instruction.

## Conditional Expressions

Conditional expressions are available in operands only. A conditional expression returns a Boolean.

A simple conditional expression consists of two operands separated by a comparison. The left operand determines the choices available in the drop-down list for the right operand. In a simple conditional expression, you compare two operands by using the following **Comparison** operators:

Comparison	Description
<	Less than
$\leq$	Less than or equal to
$\neq$	Not equal
=	Equal
>	Greater than
$\geq$	Greater than or equal

An example of a simple conditional expression is:

AdjustedRate  $\leq$  BaseRate

You can create a complex conditional expression by combining two or more conditional expressions separated by **AND** or **OR**. For example:

BaseRate  $\neq$  AdjustedRate  
AND BaseRate  $>$  10

You can add parentheses to the conditional expression. The parentheses must be balanced. For example:

( BaseRate  $\neq$  AdjustedRate )  
AND ( BaseRate  $>$  10 )

## Functions

Functions are available in operands only. A function can return a value of any type.

You can call a predefined function from a step in a rate routine. When you add a function to a step, you specify the source of the input parameters.

You can use functions for complex calculations that cannot be defined in a rate routine. For example, use a function to review the past policy's claims history and determine the experience rate factor. Or use a function to call a third-party system to compute a value. You define functions in a Gosu class.

For more information, see “Adding a New Rate Routine Function” on page 504 in the *Configuration Guide*.

## Rate Tables

Rate tables are available in operands only. The rate table returns a value of the factor or factors in the matching row of the table. The factor data types are string, integer, decimal, Boolean, and date.

You can select a rate table and return value.

1. In a rate routine step, select **Rate Table** in the **Operand** field.

PolicyCenter displays the **Select a Rate Table** screen.

2. Select a **Rate Table**.

After you select the rate table, the **Arguments** panel displays the rate table parameters. For each parameter, the **Default Source** column displays the default source of the parameter. The default source of the parameter comes from the **Argument Sources** tab of the **Rate Table Definition** screen.

3. Select a default source set from the **Argument Source Set** drop-down list. This field only appears if the rate table definition contains more than one source set. The name of the selected source set appears between curly braces. For example, the **Alternate Source** source set is selected for the **Base Rate** table:

`table:Base Rate({Alternate Source})`

4. Select a **Return Value**.

5. Optionally, specify an override to the default argument source in the **New Argument Source / Value** column.

In a multiple-factor rate table, you can select one factor or all factors. If you select all factors, the return value is a complex type composed of all the factors. For more information, see “Creating and Using a Rate Table with a Multiple Factors” on page 583.

## Constants

Constants are available in operands only. A constant can be a number, Boolean, string, or null.

A constant is the default choice for an operand. You can enter the constant in the text field. To change an operand to a constant, select **Constant** in the **Operand** field. Put quotes around a string constant. For a Boolean constant, enter `true` or `false` with no quotes. Otherwise, the constant is numeric.

## Date Constants

Date constants are available in operands only. A date constant returns a `datetime` value.

Select **Date Constant** as an operand, then choose or enter a specific date.

## TypeList Values

TypeList values are available in operands only. A typelist value is a typekey, as a string.

In an operand, you can select a value from any typelist defined in PolicyCenter. The **Value** drop-down list displays the name for each code in the typelist. Typelists are defined in Studio.

## Scale

Scale is available with rounding operators only. Returns the rounded value of the previous expression. You can set the scale to:

Value	Description
.0001	Ten thousandths
.001	Thousandths
.01	Hundredths
.1	Tenths
1	Ones
10	Tens
100	Hundreds
1000	Thousands

For more information, see Rounding Operators in “Operators in Rate Routine Steps” on page 569.

## Section Comment

A section comment is available in the **Instruction** field only. Use a section comment to provide comments about the following or preceding rows. You cannot add a section comment in the middle of a series of steps that is an

instruction. The section comment displays -- in the **Instruction** field. The comment text appears in bold and spans the remaining columns of the rate routine.

## Specifying a Function as the Operand in a Rate Routine Step

1. Navigate to an editable rate routine.

2. In a step **Operand**, select **Function....**

PolicyCenter displays the **Select a Function** screen.

3. In the **Function** field, select one of the predefined functions from the drop-down list.

In the **Arguments** panel, you select the source of each parameter value in the **Source/Value** field. The **Source/Value** field offers the same choices as selecting an **Operand**, but the selections are filtered to match the parameter **Type**. For example, if the parameter type is a string, the choices only include strings.

4. Select sources for all arguments, and click **OK** to save your work.

### See also

- “Adding a New Rate Routine Function” on page 504 in the *Configuration Guide*

## Creating a Rate Routine for Another Jurisdiction

When you create a jurisdiction variant of a rate routine, PolicyCenter creates a copy of the current rate routine with the same **Code**. For more information, see “Rate Routine Variant Identifiers” on page 541.

1. Navigate to an existing rate routine.

2. On the **Edit Rate Routine** screen, click **Create Jurisdiction Variant**. This creates a copy of the current rate routine. The copy has the same **Code**. PolicyCenter displays the **New Rate Routine** screen.

3. Select a new **Jurisdiction**. You can also change:

- **Name**
- **Description**
- **Line of Business**
- **Parameter Set**
- **Steps**

4. Make changes to the rate routine steps.

5. Click **Update** to create the copy.

6. In the **Search Results** on the **Rate Routines** screen, the **Jurisdiction** column identifies the jurisdiction variants.

The **Create Jurisdiction Variant** creates a jurisdiction variant of the current rate routine.

7. Create a rate book for this jurisdiction, and add this rate routine to that rate book. See “Adding a New Rate Book” on page 552 and “Rate Routines Included in a Rate Book” on page 554.

You can extend PolicyCenter to create variant identifiers for other features of the policy such as underwriting company. For more information, see “Configuring Variant Identifiers for a Rate Routine” on page 506 in the *Configuration Guide*.

## Specifying a Flat-rated Coverage in a Rate Routine

In a rate routine, you specify that a coverage is flat-rated rather than prorated. (Prorated is the default.) The cost of a flat-rated coverage is not prorated. For more information about costs that are flat-rated, see “Prorated or Flat Costs” on page 421.

In the sample data, the personal auto line of business contains the **Mexico Coverage - Limited** flat-rated coverage in the Personal Auto Line. You can use this example as a guide to creating your own flat-rated coverages.

#### To specify a coverage as flat-rated

1. In Product Designer, define the coverage. The definition of a flat-rate coverage is no different than any other coverage. The coverage does not have a flat-rated field.
2. In PolicyCenter, define the rate routine for the flat-rated coverage. The rate routine must have a step that sets the **ProrationMethod** instruction to the typelist value **ProrationMethod.Flat**.

The **PA Coverage Flat Rate Algorithm** rate routine provides the rating algorithm for the **Mexico Coverage - Limited** coverage. The first step of the rate routine sets the proration method to flat. The second step gets the base rate from the **Base Rate** rate table. Subsequent steps calculate the adjusted rate and term amount.

The **PA\_RTM\_Demo\_Rating** rate book contains the **Base Rate** rate table and **PA Coverage Flat Rate Algorithm** rate routine.

3. In Studio, extend the rating plugin to execute the new rate routine. This step is only necessary if you added a new rate routine. See “Configuring the Rating Engine to Execute the Rate Routine” on page 503 in the *Configuration Guide*.

In Gosu code, the coverage is rated as flat. For example, in the personal auto line, the **PARatingEngine** class executes the **pa\_cov\_flatrate** rate routine. The **PARatingEngine** class is in the **gw.lob.pa.rating** package.

The rate routine computes the properties on the cost. The **computeAmount** method in **CostData** computes the cost as a flat cost because the rate routine set the proration method to flat. The **CostData** class is in the **gw.lob.pa.rating** package.

## Viewing Rating Worksheets

The user must have the **View rating worksheet** permission to view a rating worksheet. The code for this permission is **ratingworksheetview**.

1. In PolicyCenter, navigate to the **Quote** screen of a policy or policy transaction.

The **Show Rating Worksheet** button appears on the **Policy Premium** tab on the **Quote** screen of policy or policy transaction.

2. Click **Show Rating Worksheet** to view the **Rating Worksheets** popup window.

3. Expand a coverage or other rated item to view:

- Rate book code and edition
- Rate routine code and version
- The **Result** column displays the result of each step in the rate routine.
- The **Operand Value** column displays the value of the operand at the time the rate routine was executed.

4. Click **Show Conditionals** to show conditional instructions and conditional expressions. By default, the rating worksheet does not show conditional instructions or expressions. Click this button to show that a particular **IF** statement was chosen and also show the **ELSE** path not taken. Click this button to show the values in a conditional expression such as **BaseRate < MinimumPremium**.

#### See also

- “Rating Worksheets” on page 548

## Working with Parameter Sets

In PolicyCenter, you can associate parameter sets with rate tables and rate routines. Parameter sets define the contextual information that is passed to rate routines. In rate tables, you specify the argument source for each

parameter in the parameter set. This topic describes how to work with parameter sets in the PolicyCenter user interface.

## Accessing Parameter Sets

1. In PolicyCenter, select **Administration** → **Rating** → **Parameter Sets**.

PolicyCenter displays the **Parameter Sets** screen. By default, the screen displays the parameter sets for all lines.

2. From the **Policy Line** drop-down list, select a policy line such as **Personal Auto Line**.

PolicyCenter displays the parameter sets defined for the selected policy line.

3. Select a parameter set to display its parameters in the **Parameters** tab.

## Adding a Parameter Set

1. On the **Parameter Sets** screen for a policy line, click **Add**.

PolicyCenter adds a blank parameter set to the list of parameter sets.

2. Enter values for the **Name** and **Code** of the parameter set.

## Adding, Changing, or Removing Parameters from a Parameter Set

1. In PolicyCenter, navigate to **Administration** → **Rating** → **Parameter Sets** screen.

2. Select a parameter set.

PolicyCenter displays the parameters in that parameter set in the **Parameters** tab.

3. In the **Parameters** tab, click **Edit** to edit the selected parameter set.

The **Edit** button appears if:

- A rate table definition or a rate routine included in a rate book with **Draft** status uses the parameter set.
- No rate table definition or rate routine uses the parameter set.
- No rate book includes a rate table definition or rate routine that includes the parameter set.

4. In the **Parameters** tab, click **Add** to add a parameter.

5. Specify the following information for the parameter:

Field	Description
<b>Name</b>	Select a name from the drop-down list. The names are specified in Studio.
<b>Type</b>	A Gosu expression which describes the data type for this parameter. When you insert the parameter, this value is set to the default type. You can override this value.
<b>Coverage</b>	If the <b>Type</b> specifies an entity that is a coverage, then you can select a specific coverage in the policy line. For example, a parameter with <b>Type</b> set to <code>entity.Coverage</code> specifies an entity that is a coverage.
<b>Writable</b>	Select this field if the parameter's properties can be overwritten. By default, the parameter is not writable. If a parameter is writable, you can select its properties in the <b>Instruction</b> field of a step.

In some cases, it is necessary to override a parameter's default **Type**. For example, the **PolicyLine** parameter has a default of `entity.PolicyLine`. When you include this parameter in a line-specific parameter set, you override the default type with the type for that parameter set. For personal auto, set the parameter's **Type** to `entity.PersonalAutoLine`.

You can define a parameter that provides a rate routine access to coverage terms, options, and packages in policy data. Define a parameter with the **Type** field specifying `entity.Coverage`, and the **Coverage** field speci-

fying a coverage pattern code. For more information, see “Accessing Coverages in Parameters” on page 573.

6. To edit a parameter, modify the parameter definition directly in the **Parameters** tab.

You cannot edit a parameter if a rate table definition or rate routine step references that parameter.

7. To delete a parameter, add a check mark in the first column of the parameter definition, then click **Delete**.

You cannot delete a parameter if a rate table definition or rate routine step references that parameter.

8. Click **Update** to save your work.

#### See also

- “Filtering Parameter Set Fields of the Same Type When Entering Parameters in a Rate Routine” on page 572
- “Configuring New Parameters in Parameter Sets” on page 511 in the *Configuration Guide*

## Adding Policy Line Rate Modifiers to a Parameter Set

Rate modifiers on the policy line are accessible through **PolicyLine** parameter. In the base configuration, the **PolicyLine** parameter default type is `entity.PolicyLine`. To make these rate modifiers accessible through the parameter set, you can add a parameter with the following values:

- **Name** – `PolicyLine`
- **Type** – `entity.JobLine` as a fully qualified object type. For example, enter `entity.CommercialPropertyLine` for the commercial property line.

#### See also

- “Parameters” on page 572 in the *Application Guide*

## Working with Impact Testing

You can use impact testing to see the impact that changing the rate book has on policy premium for a group of policies. Impact testing generates test policy periods and rates them using the active rate books and selected comparison rate books. You can choose the test policy periods by product, jurisdiction, producer code, postal code, effective date, expiration date, and in force on date.

### Impact Testing Warnings and Recommendations

Impact testing is designed to work in a test environment on a copy of production data. Impact testing is accessible only when the server is in development or test mode. Because impact testing affects system performance and creates test policy periods that persist in the database, impact testing is not accessible in production mode.

When you run impact testing, impact testing creates test policy periods in the database. In the test environment, observe the following warnings and recommendations:

- The work of users can interfere with impact testing results. During impact testing, Guidewire recommends that only the single user performing impact testing be logged into PolicyCenter.
- Disable integrations to other systems. If you run impact testing with integrations enabled, the integration can send test data to these production systems. For example, disable the free-text search integration that uses the search engine Solr. The same applies to other integrations such as an integration with a billing system. To avoid unnecessary costs, disable integrations to systems that charge access fees.
- While running impact testing batch processes, do not make changes in PolicyCenter that impact rating such as doing a policy change, canceling, or changing rate books. For example, the baseline creation batch process or the test period quote batch process is running. During this time, quoting test periods fails if a user changes or cancels a policy which has a baseline.
- Do not include expired policies in the impact testing dataset. Specify an **In Force On** date to filter out expired policies.

- Impact testing excludes archived policies.
- Stop the Job Expire batch process to prevent PolicyCenter from unexpectedly expiring policies during or between impact testing runs. To manage batch processes on a test server, press ALT + SHIFT + T to display the **Server Tools** page, then select **Batch Process Info** from the **Server Tools** tab.

#### See also

- “Impact Testing” on page 549
- “Impact Testing Plugin” on page 521 in the *Configuration Guide*
- “Work Queues for Impact Testing” on page 517 in the *Configuration Guide*

## Permissions for Impact Testing

To view the **Administration** → **Rating** → **Impact Testing** screen, you must have the **Rate policies for rate impact testing** permission. The code for this permission is `rateimpacttesting`. In the base configuration, the **Rating Analyst** and **Super-user** roles have this permission.

If a user has this permission, impact testing analyzes and provides access to all policy periods that match the search criteria. With this permission, the user can access policy periods for which the user has insufficient permissions.

## Generating Test Periods with Impact Testing

This topic provides step-by-step instructions on how to use impact testing to generate test policy periods rated using the current rate books and rate books that you select.

### Impact Testing Warnings and Recommendations

Impact testing is designed to work in a test environment on a copy of production data. Impact testing is accessible only when the server is in development or test mode. Because impact testing affects system performance and creates test policy periods that persist in the database, impact testing is not accessible in production mode.

When you run impact testing, impact testing creates test policy periods in the database. In the test environment, observe the following warnings and recommendations:

- The work of users can interfere with impact testing results. During impact testing, Guidewire recommends that only the single user performing impact testing be logged into PolicyCenter.
- Disable integrations to other systems. If you run impact testing with integrations enabled, the integration can send test data to these production systems. For example, disable the free-text search integration that uses the search engine Solr. The same applies to other integrations such as an integration with a billing system. To avoid unnecessary costs, disable integrations to systems that charge access fees.
- While running impact testing batch processes, do not make changes in PolicyCenter that impact rating such as doing a policy change, canceling, or changing rate books. For example, the baseline creation batch process or the test period quote batch process is running. During this time, quoting test periods fails if a user changes or cancels a policy which has a baseline.
- Do not include expired policies in the impact testing dataset. Specify an **In Force On** date to filter out expired policies.
- Impact testing excludes archived policies.

- Stop the Job Expire batch process to prevent PolicyCenter from unexpectedly expiring policies during or between impact testing runs. To manage batch processes on a test server, press ALT + SHIFT + T to display the Server Tools page, then select Batch Process Info from the Server Tools tab.

**Note:** Impact testing has several steps that may require a long time to complete. For these time-consuming steps, impact testing uses batch processes. While PolicyCenter is generating results, the user can navigate away from impact testing screens. The user can return later to view results and continue a previously started test. The following instructions include steps for continuing a previously started test.

#### To generate test periods

- In PolicyCenter, navigate to Administration → Rating → Impact Testing.  
PolicyCenter displays the Choose Policies screen.
- Do one of the following:
  - To continue a previously started test, click Next with Previous Test Case to advance the wizard. Go to step 6.
  - Specify criteria to filter policies for testing. None of these fields is required.

Field	Description
Products	Choose one or more products.
Jurisdictions	Choose one or more jurisdictions.
Producer Codes	Choose one or more producer codes.
Postal Codes	Choose one of the following: <ul style="list-style-type: none"><li>Specify codes – One or more postal codes separated by commas.</li><li>Range – Specify the minimum and maximum for a range of postal codes. The minimum and maximum are inclusive. PolicyCenter searches for a match that starts with the minimum and maximum that you enter.</li></ul>
Effective Date	The policy effective date must fall within this range.
Expiration Date	The policy expiration date must fall within this range.
In force on	Policies must be in force on this date to be included in the search results.

- Click Search to display the list of policies that meet the search criteria.
- Repeat specifying criteria in step 2 and search in step 3 until you are satisfied with the search results.
- Click Next with Search Results. Alternatively, you can choose Next with Previous Test Case to continue and use the previous search results.  
PolicyCenter displays the Create Baseline screen.
- Depending upon the button that you clicked in step 5, do one of the following:
  - Next with Search Results – Click Create Baseline to generate baseline policy periods on the policies in the search results. The policy periods are rated using the current rate books.  
For all policy transactions, impact testing rates the baseline policy periods as submission policy transactions. For information on configuring impact testing to rate renewals as renewal policy transactions, see “Rating Renewals as Renewal Jobs in Impact Testing” on page 516 in the Configuration Guide.  
When you click Create Baseline, PolicyCenter runs the Impact Testing Test Case Preparation batch process. Since this batch process may take a long time to run, you may exit impact testing, and go to other PolicyCenter screens. After the Impact Testing Test Case Preparation batch process finishes creating baselines, the Create Baseline screen displays the baseline policy periods for each policy in the search results.  
After the batch process completes, the Next button appears.
  - Next with Previous Test Case – If you are continuing a previously started test, the baselines are already created and appear on the Create Baseline screen.

The **Create Baseline** screen displays the baseline policy periods for each policy in the search results. The first column displays:

- A check mark if the baseline was successfully created
- An X if the baseline creation failed. Click the X to view more information about why baseline creation failed.

The **Baseline Period** column contains the submission number of the policy transaction that created the baseline policy period. Click this link to display the **Impact Testing Policy Period Overview** screen. This screen contains information about the baseline and test policy periods. The screen displays a baseline overview which includes premium details.

7. If you arrive at the **Create Baseline** screen by clicking **Next with Previous Test Case**, the screen has the following additional buttons:
  - Click **Reprocess Failures** to reprocess all test periods that failed to generate baseline policy periods during the previous run.
  - Click **Recreate Baseline** to regenerate all baseline policy periods.
8. Click **Next**. PolicyCenter displays the **Select Rate Books** screen. The rate books are grouped into **Available Active Rate Books** and **Available Stage or Approved Rate Books** panels. This screen only displays rate books from the policy lines contained in the selected products.
9. Select one or more rate books to move them to **Selected Rate Books** panel.

If you select more than one rate book, PolicyCenter rates the policy using the rate book with matching jurisdiction, underwriting company, policy line, and other factors. If more than one rate books applies, then PolicyCenter rates the test periods using the most recently changed rate book. When selecting a rate book for the test periods, PolicyCenter does not consider the rate book effective date.
10. Click **Next** to advance to the **Testing Periods** screen.

Similar to baseline policy periods, impact testing rates all test policy periods as submission policy transactions. For information on configuring impact testing to rate renewals as renewal policy transactions, see “Rating Renewals as Renewal Jobs in Impact Testing” on page 516 in the *Configuration Guide*.
11. Do one of the following:
  - To continue a previously started test in which the test periods are already quoted, click **Requote Test Periods** or go to step 12.
  - Click **Quote Test Periods** to generate test policy periods rated using the selected rate books. When you click this button, PolicyCenter runs the **Impact Testing Test Case Run** batch process. You may exit impact testing and return to this screen later.

If you return to the **Select Rate Books** screen, the list of rate periods includes a **Test Period** column. If an applicable rate book was found, the \$ indicates that the policy period was rated.

When the batch process completes, the **Next** button appears.
12. Click **Next** to advance to the **Impact Results** screen.

This screen displays **Policies Affected** and **Financial Impact** bar graphs. In both graphs, the X-axis is divided into **Percent Change** ranges. Each range represents a percentage change to the policy premium, such as 0 for no change or >5 for a 0% up to 5% change. In the **Policies Affected** graph for each bar, the Y-axis represents the number of policies in the impact range. In the **Financial Impact** graph for each bar, the Y-axis represents monetary amount of change in each impact range.

A table between the graphs displays the following information for each range:

- # – Number of policies affected in the impact range. This is the same information that the **Policies Affected** graph shows.
- % – Percentage of policies in the impact range.
- \$ – Monetary amount of change in each impact range. This is the same information that the **Financial Impact** graph shows.

**13.** Click **Create Excel Export File** to export the test periods to Microsoft Excel format. When you click this button, PolicyCenter runs the **Impact Testing Export** batch process. You may exit impact testing and return to this screen later.

**14.** Click **Download Excel Export File** to open or save the results.

For each policy the Excel spreadsheet shows details for each cost on the policy when rated using the active rate book and the comparison rate book. The cost details include:

**Baseline Rate Book**

- **Baseline Term Amount**
- **Baseline Actual Amount**

**Comparison Rate Book**

- **Comparison Term Amount**
- **Comparison Actual Amount**
- **Errors** – Displays an error message if no rate book matched, or rating generated an invalid quote or encountered some other problem. The term amount and actual amount columns are blank.

## Examples of Working with Rating Management

This topic provides examples that access multiple features of Rating Management. The examples assume that you have loaded the small sample data set.

- “Creating and Using a Rate Table with a Multiple Factors” on page 583

### Creating and Using a Rate Table with a Multiple Factors

These step-by-step instructions show you how to create a rate table with multiple factors, then access the rate table factors in rate routine steps.

**See also**

- “Rate Table with Multiple Factors” on page 538

#### Creating a Rate Table with Multiple Rate Factors

In this topic, you add two additional rate factors to a copy of the **Base Rate** rate table.

1. In PolicyCenter, select **Administration** → **Rating** → **Rate Table Definitions**.
2. On the **Rate Table Definitions** screen, select **Commercial Property Line** from the **Policy Line** drop-down list and click **Search**.
3. In the **Search Results**, click **Base Rate** to view the rate table.
4. Click the **Factors** tab.

This table has one factor named **Base Rate**.

5. Click **Copy** to create a copy of this rate table and enter the following on the **Basics** tab of the new rate table:

Field	Value
Code	BASE_RATE_MULTIPLE_FACTORS
Name	Base Rate with Multiple Factors

6. On the **Factors** tab, add a decimal factor with the following values on the **Factor Details** tab:

Field	Value	Additional information
Data Type	Decimal	
Decimal Places	3	The number of decimal places the user can enter in the rate factor row. This field only appears for the Decimal data type.
Priority	10	
Column Code	Integer Factor	
Column Label	Integer Factor	
Display Type	Normal	
Physical Column	Select a column such as dec1 (15, 7)	This decimal value has a precision of 15, and a scale of 7. The decimal value can have 15 digits, including 7 decimal places.

7. On the **Factors** tab, add a Boolean factor with the following values on the **Factor Details** tab:

Field	Value
Data Type	Boolean
Priority	20
Column Code	Boolean Factor
Column Label	Boolean Factor
Display Type	Normal
Physical Column	bit1

8. Click **Update** to save your changes.

## Using the Multiple Factor Rate Table in a Rate Routine

- In PolicyCenter, select **Administration** → **Rating** → **Rate Routines**.
- On the **Rate Routines** screen, click **Add** to create a new rate routine.
- Enter the following values on the **New Rate Routine** screen:

Field	Value
Code	CP_MULTIPLE_FACTOR_RR
Name	CP with Multiple Factor
Line of Business	Commercial Property Line
Parameter Set	CP Coverage Parameter Set

Next, you add a step that creates a variable named **varMultiFactor** that contains all the factors from the rate table.

- In the **Instruction** field of the first step, select **New** → **New Variable** and name it **varMultiFactor**.
- Select  $\leftarrow$  as the **Op**.
- In the **Operand**, select **Rate Table**, and select the **Base Rate with Multiple Factors** rate table.
- For **Return Value**, select **[All]**.
- Specify **New Argument Source / Value** for each parameter, then click **OK**.

Next, you add several steps that make up one instruction. The steps access individual factors on the multiple factors variable.

9. Select **Append → Add 10 Rows**.
10. In the **Instruction** field of the next step, select **Base Rate**.
11. Select  $\leftarrow$  as the **Op**.
12. In the **Operand**, select **Variable → varMultiFactor [Base Rate]**.
13. In the **Instruction** field of the next step, select **IF**.
14. In the **Operand**, select **Variable → varMultiFactor [Boolean Factor]**.

Continue adding steps to the rate routine.



---

part VII

# Reinsurance Management



# Reinsurance Management Concepts

Reinsurance is insurance risk transferred to another insurance company for all or part of an assumed liability. Reinsurance can be thought of as insurance for insurance companies. When a company reinsurance its liability with another company, it cedes business to that company. The amount an insurer keeps for its own account is its retention. When an insurance company or a reinsurance company accepts part of another company's business, it assumes risk. It thus becomes a reinsurer.

**Note:** The insurance company directly selling the policy is also known in the industry as the carrier, the reinsured, or the ceding company. This topic uses the term *carrier* to refer to this company. An insurance company accepting ceded risks is known as the *reinsurer*.

Guidewire Reinsurance Management provides reinsurance for all lines of business.

---

**IMPORTANT** Guidewire Reinsurance Management is available within Guidewire PolicyCenter. To determine whether your Guidewire PolicyCenter license agreement includes Reinsurance Management, contact your Guidewire sales representative. Reinsurance Management requires an additional license key. For instructions on obtaining and installing this key, contact your Guidewire support representative.

---

This topic includes:

- “Reinsurance Program Basics” on page 590
- “Reinsurance Agreements” on page 595
- “How PolicyCenter Links Reinsurance to Policies” on page 604
- “How PolicyCenter Calculates Ceded Premiums” on page 607
- “Shared Reinsurance Agreements” on page 610
- “Location Groups” on page 611

**See also**

- “Multicurrency and Reinsurance” on page 521

## Reinsurance Program Basics

Carriers procure reinsurance in the form of treaties and facultative agreements. A facultative agreement is a reinsurance agreement for a specific risk that is negotiated on an individual case basis. A treaty, on the other hand, is an agreement between the carrier and the reinsurer to provide coverage for all risks of a certain type.

Carriers group reinsurance treaties into reinsurance programs to cover policy risks in a way that maximizes their business goals. Carriers also group treaties into programs to ensure that they have no gaps in coverage and to ensure that they do not duplicate coverage.

A carrier typically operates several reinsurance programs. The carrier structures each reinsurance program to cover a class of risks within a monetary range. Risks that are large and rare are not usually covered by treaties in a reinsurance program. Facultative agreements handle these risks.

After the reinsurance programs are set up in PolicyCenter, PolicyCenter attaches each qualifying policy risk across all the carrier's open policies to one and only one active reinsurance program. At this point, PolicyCenter also attaches all the treaties of that program both to the policy and to that specific policy risk.

On the policy level, a policy held by the carrier can be associated with several programs, one for each class of risks. The policy can also be attached to one or more treaties within each program.

Within PolicyCenter, these attachments of program to policy risk are then used to:

- Calculate ceded premiums.
- Calculate proportional shares on exposures for export to claims systems, such as ClaimCenter.

Within a claim system, such as ClaimCenter, these attachments can be retrieved to display the following reinsurance information associated with a claim or set of claims:

- Treaties that are associated with a specific exposure
- How exposures from one or more claims are grouped into a single reinsurance risk
- The amount that can be recovered from each reinsurer on each policy risk, itemized on a per exposure basis

PolicyCenter additionally enables creating and applying facultative agreements against individual high risks that are not covered by any of the PolicyCenter reinsurance program treaties.

## Using Reinsurance Programs to Serve Business Goals

A carrier might want to transfer their risk of loss for several reasons:

- To protect capital and maintain solvency
- To provide a more even flow of net income over time by flattening out claims losses
- To take on more business and across a larger set of risks than the carrier would normally retain
- To spread risk over the globe and take advantage of currency advantages
- To provide catastrophe relief
- To withdraw from a line of business

The carrier might find it advantageous to bundle various types of reinsurance in a way that maximizes its ability to achieve these business goals.

For instance:

- Carriers that want to increase capacity benefit from reinsurance that either takes a percent of the risk or takes a loss above a certain point. If a carrier can be free of fear of multiple large losses, it can comfortably take on more risk.
- Carriers that seek to stabilize their net income flow benefit from reinsurance that takes a percent of the loss above a certain point.

- Carriers that want to withdraw from a line of business benefit from reinsurance that takes on a percentage of risk under a certain loss point for that line of business.

Whether a carrier has one or more of these business goals in mind, common industry practice has established that the carrier can achieve these goals through reinsurance. In setting up reinsurance programs, carriers take into account factors such as:

- The carrier's average policy claim losses and premium intake
- Likelihood of catastrophe
- Proximity of policies taken out in a geographic location

Carriers group reinsurance treaties into reinsurance programs to cover policy risks in a way that maximizes their business goals. They also group treaties into programs to ensure that they have no gaps in coverage and to ensure that they do not duplicate coverage.

## Reinsurance Program Design

A reinsurance *program* is a set of reinsurance treaties designed to insure policy risks for all policies held by the carrier that fall:

- Within one type of line of business or peril.
- Under a certain monetary cap.

The line of business or peril covered by the reinsurance program is also known as the *reinsurance coverage group*.

Each reinsurance program is typically active for one year, and reinsurers that participate in the reinsurance program are bound to automatically cover qualifying claim losses.

Carriers typically assemble one reinsurance program per reinsurance coverage group.

Part of reinsurance design is deciding what types of risks are too expensive to include wholesale into the reinsurance program. If a risk on any policy falls outside the risks covered by the carrier's reinsurance programs, the carrier can take out a facultative reinsurance agreement for that particular risk. Often, the carrier and the reinsurer both have the option to act or not act on this facultative agreement when a qualifying loss occurs.

Sometimes, a set of individual risks that are deemed as individual facultative risks can be bundled into a treaty rather than be handled by a series of individual facultative agreements. This type of treaty is known as a *hybrid*. The reinsurance program can include hybrid treaties.

## Risk in the Reinsurance Program

The base unit of reinsurance in a reinsurance treaty is a risk, not the coverable. Often, several coverables are grouped into one risk. For instance, a policy might have a location that includes two buildings. For this policy, the buildings are the coverables, and the location is the risk. If a claim has only one exposure against one building, the total insured value of the location determines the share paid by each reinsurer.

## Reinsurance Coverage Group and Reinsurance Program Risk Type

A commercial property carrier typically has one active reinsurance program for policy risks falling in the property reinsurance coverage group. The carrier typically also has another active reinsurance program for policy risks falling in the liability reinsurance coverage group. The carrier might also have one future and multiple past programs for each of these reinsurance coverage groups.

In Policy Center, the default configuration contains the following reinsurance coverage groups:

- Auto Liability
- Auto PD
- Liability

- Property
- Workers' Compensation

**See also**

- “Reinsurance Coverage Groups for Treaties” on page 595

## Treaties and Program Monetary Layers in a Reinsurance Program

Policy risks of one reinsurance coverage group fall into various monetary ranges, and different reinsurers for that particular reinsurance coverage group cover some monetary ranges more aptly than others.

To build a reinsurance program, the carrier assembles one or more reinsurance treaties with the same reinsurance coverage type. Each treaty provides a different type of risk or loss coverage and provides it for a different monetary layer or range than the other treaties. These various treaties are arranged in the program to yield a measurable business advantage.

Each individual treaty can be drawn up with a different reinsurer from the other treaties. In addition, each individual treaty covers one and only one of the following:

- A different layer of monetary risk against all policies that have coverables in that reinsurance coverage group
- A different monetary range of loss for qualifying risks above a certain attachment point and below a cap

Treaties that provide coverage based on the risk are broadly known as *proportional treaties*. Treaties that provide coverage against loss for qualifying monetary risks between an attachment point and a cap are known broadly as *non-proportional treaties*, or *excess of loss* treaties.

**See also**

- “Treaties” on page 595.
- “Proportional Agreements” on page 596.
- “Non-proportional Agreements” on page 599.

## Using Facultative Agreements in Addition to Programs

After a carrier puts its reinsurance programs into place, there still might be some remaining risks that were too expensive to insure within the treaty framework. In addition to programs, the carrier can add facultative reinsurance agreements to cover the uncovered portion of each very high risk.

For more information, see “Facultative Agreements” on page 596.

## Reinsurance Program Example

The reinsurance program example in this topic illustrates the reinsurance program concepts explained in “Reinsurance Program Design” on page 591.

In our example, the reinsurance carrier operates two reinsurance programs: one for property risks and one for liability risks.

Each topic in our example details a different aspect of the example reinsurance program structure, specifically:

- “Risk and Loss in Treaty Composition” on page 593.
- “Risks and Coverables When Applying Program Treaties to a Loss” on page 593.
- “Attaching Policies to Reinsurance Programs and Treaties” on page 594.

## Risk and Loss in Treaty Composition

### Property Reinsurance Program

The carrier's property reinsurance program provides two types of reinsurance:

- Reinsurance up to \$5 million per risk for all the policies that have coverables in the property reinsurance coverage group.
- Reinsurance against losses incurred on a per risk basis for all policies that have coverables in the property reinsurance coverage group from \$5 million to \$10 million.

The property reinsurance program contains three treaties:

Property Treaty	Description
Treaty Three: Excess of Loss	Provides reinsurance on losses from \$5 million to \$10 million.
Treaty Two: Proportional	Provides reinsurance based on policy risk from \$1 million to \$5 million.
Treaty One: Proportional	Provides reinsurance based on policy risk up to \$1 million.

This property program enables the carrier to take in more business below \$5 million than it had in the past. The carrier can take in more business because the carrier shares the risk of loss payments with two other reinsurers under Treaty One and Treaty Two.

Treaty Three enables the carrier to expand into a higher risk, more lucrative market. Before adding Treaty Three, the carrier could not afford the risk of insuring losses above \$5 million. The carrier finds it attractive to include Treaty Three in its property program because losses above \$5 million occur infrequently. The carrier can enjoy the net income generated by the larger policy premiums while paying a relatively low premium to cover these rarer loss types.

### Liability Reinsurance Program

The liability reinsurance program provides the carrier reinsurance at the policy line level. Treaty Four is written to cover overall liability risks up to \$20 million, and Treaty Five is written to cover the risk layer from \$20 million to \$50 million. This type of program allows the carrier to insure higher risk locations by reinsuring the rarer large liability losses without ceding premiums.

Liability Treaty	Description
Treaty Five: Excess of Loss	Provides reinsurance based on policy risk from \$20 million to \$50 million.
Treaty Four: Proportional	Provides reinsurance based on policy risk up to \$20 million.

## Risks and Coverables When Applying Program Treaties to a Loss

A commercial property policy holder suffers a loss on a building at one location and another loss at a building in a different location. Each building is a coverable. The risk associated with each building is calculated based on the sum of all coverables at its location, which is otherwise known as a location group.

The total insured value of the risk is used to decide which treaties share in a loss against any of the coverables in the risk. The loss amount is used to determine the actual percentage by which each of the applied treaties shares in the loss.

For example, the policy holder has a Location One with a total insured value of \$8 million. When the policy is drawn up, Treaty One, Treaty Two, and Treaty Three from the carrier's property program are attached to this policy risk. These treaties are attached because these policies fall within the risk amount covered by these treaties. But Treaty Three only applies up to \$8 million. This program is described in "Risk and Loss in Treaty Composition" on page 593.

The policy holder then suffers a \$6 million loss at Location One. Treaties One and Two share in this loss from zero, and their share is calculated by multiplying their percentage of loss multiplied by the amount that they reinsurance. This share is then divided by the total insured value. Treaty Three pays a simple \$1 million.

Then the policy holder suffers a loss of \$500,000 on a single building at Location Two, which has two buildings with a total insured value of \$4 million. Since the total risk is \$4 million, the insurer uses only Treaty One and Treaty Two to recover loss from the reinsurers. The proportion that the two reinsurers share in the loss is otherwise known as the proportional share. Detailed examples of proportion determination are provided in “Example of Ceding Risk to Proportional Treaties” on page 596.

The policy holder in our example has a liability risk over their two locations of \$15 million. When the policy holder suffers loss due to building damage at Location One, several persons suffer physical harm. At a later date, the building at Location Two incurs damage, and people at that building also undergo physical harm.

In both these cases, only Treaty Four from the liability reinsurance program applies. Additionally, the proportion in which each reinsurer participates is calculated solely on the basis of the current loss at each location.

### Attaching Policies to Reinsurance Programs and Treaties

In the preceding property example, Treaty One and Treaty Two share with the carrier in any loss beginning from zero on any risk that falls within their covered risk amount. For example, the carrier recovers from both Treaty One and Treaty Two on any property risk with a total insured value of \$1.5 million.

The carrier recovers from Treaty Three only when the following are both true:

- There is a claim against a policy risk that has a total insured value greater than \$5 million
- The loss is greater than \$5 million.

The carrier can recover only up to the smaller of the total insured value or \$10 million.

In short, all of the carrier’s policies attach to the treaties in this property reinsurance program as follows:

- Treaty One covers all policies with property risks that have total insured value up to \$1 million.
- Treaties One and Two cover all policies with property risks that have a total insured value up to \$5 million.
- All three treaties cover all policies with property risks that have a total insured value more than \$5 million. While Treaties One and Two begin to apply from the first dollar on any loss, Treaty Three begins to apply only when the loss is greater than \$5 million. The maximum loss coverage provided by Treaty Three is \$10 million.

The liability program and its treaties attach to policies in a similar manner:

- Treaty Four covers all policies with liability risks that have a total insured value up to \$20 million.
- Treaties Four and Five cover all policies with liability risks that have a total insured value on proper risk up to \$50 million.

Policies can attach to the programs and treaties in various ways:

- Some policies only attach to the property program. A policy attaches only to those treaties that fall within the monetary range of its risk. So for some policies this will be only Treaty One. For others it will be Treaty One and Treaty Two. For yet others it will be Treaty One, Treaty Two, and Treaty Three.
- Some policies only attach to the liability program, but a policy will attach only to those treaties that fall within the monetary range of its risk. So for some this will be only Treaty Four, and for others it will be Treaty Four and Treaty Five.
- Some policies attach to both programs, but a policy only attaches to those treaties that fall within the monetary range of its risk in each program. For example, a policy might attach only to Treaty One in the property reinsurance program, but attach to both Treaties Four and Five in the liability reinsurance program.

### See also

“How PolicyCenter Links Reinsurance to Policies” on page 604

## Reinsurance Agreements

There are two kinds of reinsurance agreements, *treaties* and  *facultative agreements*.

- **Treaty** – An agreement between the carrier and the reinsurer to provide coverage for all risks of a certain type.
- **Facultative agreement** – An agreement for a specific risk that is negotiated on an individual case basis.

Each of these agreement types can be drawn up as either a proportional or a non-proportional agreement. Proportional and non-proportional agreements share the risk, premium, and payment for loss with the reinsurer in different ways:

- **Proportional Reinsurance** – Transfers a percentage of the risk to the reinsurer. The reinsurer receives that percentage of the premium and is responsible for that percentage of each loss. Proportional reinsurance is always per risk coverage—it covers one risk.
- **Non-proportional Reinsurance** – There is no proportional ceding of the risk and no proportional sharing of the premium or the losses. The carrier pays the entire loss up to an agreed amount called the attachment point. The reinsurer pays all or part of the loss that exceeds the attachment point up to a limit previously agreed upon by the carrier and reinsurer.

This topic includes:

- “Treaties” on page 595
- “Facultative Agreements” on page 596
- “Proportional Agreements” on page 596
- “Non-proportional Agreements” on page 599
- “Summary of Agreement Types” on page 603

### Treaties

A treaty is an agreement between the carrier and the reinsurer that provides reinsurance without the carrier having to submit every risk to the reinsurer. The treaty is a contract, usually arranged on a yearly basis, that covers a class of risks for a monetary range of total insured value. The carrier cedes to the reinsurer a portion of each risk that the treaty covers.

For example, the carrier has a treaty with a reinsurance company. The reinsurance company agrees to pay 40% of property damage claims when the claim amount is between \$1 million and \$5 million.

#### See also

- “Proportional Treaties” on page 596
- “Non-proportional Treaties” on page 600

### Reinsurance Coverage Groups for Treaties

You can group individual coverages into a reinsurance coverage group. Treaties are written to cover the total loss against a broad category of coverages. For example, a reinsurance group might contain coverages for building, contents, and business interruption. A treaty provides coverage for one or more of these reinsurance coverage groups.

In PolicyCenter, the default configuration contains the following reinsurance coverage groups:

- Auto Liability
- Auto PD
- Liability
- Property
- Workers’ Compensation

## Facultative Agreements

Facultative agreements are always for per risk insurance. They are used to reinsure risks that do not fall within the reinsurance coverages provided by the treaties in a program.

For a specific risk, the carrier and the reinsurer each have free choice in arranging the reinsurance. The carrier is free to decide whether or not to reinsure a particular risk and can offer the reinsurance to any reinsurer it chooses. By the same token, it is at the reinsurer's discretion whether to accept any risk offered, decline it, or negotiate different terms.

A facultative agreement provides reinsurance for claims that fall within a specified range. The facultative agreement reinsures a specific amount.

For example, a policy provides insurance up to \$4 million. A number of treaties provide coverage for claims up to \$2 million. For a specific risk on the policy, the carrier negotiates two proportional facultative agreements to provide coverage for claims valued at \$2 million to \$4 million. One facultative agreement provides reinsurance coverage for \$500,000. The second facultative agreement provides reinsurance coverage for \$1.5 million. If the risk suffers a loss of \$4 million, the treaties provide reinsurance for the first \$2 million. The two facultative agreements provide reinsurance for the remaining \$2 million.

### See also

- “Proportional Facultative Agreements” on page 598
- “Non-proportional Facultative Agreements” on page 602

## Proportional Agreements

Reinsurance Management provides proportional reinsurance for both treaties and facultative agreements.

Proportional reinsurance transfers a percentage of the risk to the reinsurer. The reinsurer receives that percentage of the premium and is responsible for that percentage of each loss. Proportional reinsurance is always *per risk* coverage—it covers one risk.

### Proportional Treaties

Reinsurance Management provides two types of proportional treaties:

- **Quota share** – The reinsurer assumes an agreed-upon percentage of each relevant risk and shares all premiums and losses accordingly with the reinsured. For example, a carrier has a 40% quota share on all homeowners policies. For every policy, 40% of the premium is ceded to the reinsurer. The reinsurer is responsible to pay for 40% of all losses. A quota share treaty provides reinsurance coverage starting at \$0 up to a coverage limit.
- **Surplus** – The surplus treaty provides reinsurance coverage from a starting value up to the coverage limit. The way in which the percentage of premium is ceded and losses are paid is similar to quota share.

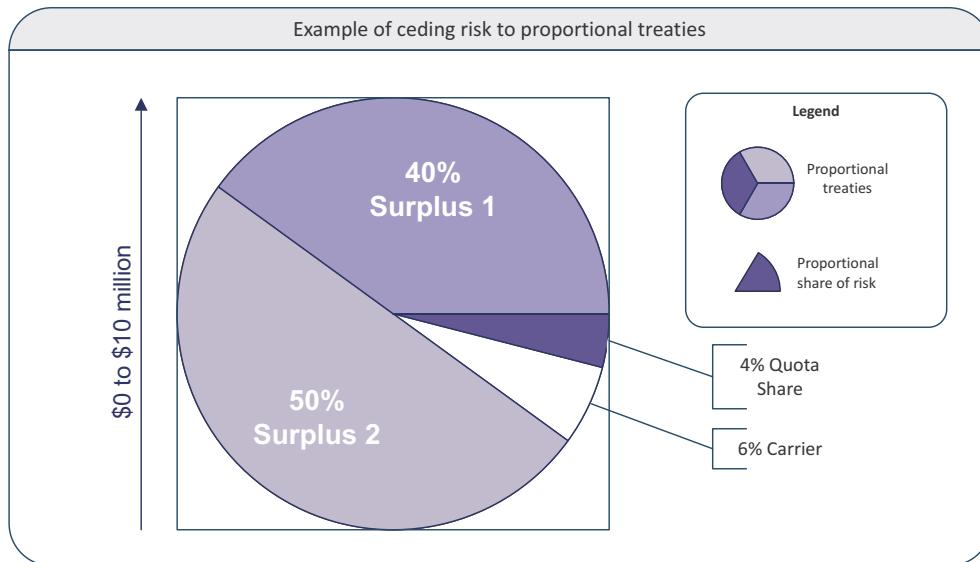
### Example of Ceding Risk to Proportional Treaties

In a reinsurance program, quota share and surplus treaties provide layers of reinsurance coverage. In the following example, three proportional treaties provide reinsurance coverage up to \$10 million:

Treaty	Layers of reinsurance	Monetary risk ceded to reinsurer	Proportional share of risk
Surplus 2	From \$5 million to \$10 million	\$5 million	\$5 million of \$10 million = 50%
Surplus 1	From \$1 million to \$5 million	\$4 million	\$4 million of \$10 million = 40%

Treaty	Layers of reinsurance	Monetary risk ceded to reinsurer	Proportional share of risk
Quota share	From \$0 to \$1 million ceding 40% of the risk to the reinsurer	\$400,000	\$400,000 of \$10 million = 4%
Carrier's share	From \$0 to \$1 million 60% of the risk retained by the carrier	\$600,000	\$600,000 of \$10 million = 6%

The treaties share a \$10 million risk proportionally as shown in the following illustration:



When there is a loss of \$10 million or less on a risk with a total insured value of \$10 million, the proportional treaties share the loss proportionally. The amount of each treaty's share is shown in the last two columns of the following table:

Treaty	Proportional share of loss	\$10 million loss	\$5 million loss
Surplus 2	50% of loss amount	\$5 million	\$2.5 million
Surplus 1	40% of loss amount	\$4 million	\$2 million
Quota share	4% of loss amount	\$400,000	\$200,000
Carrier's share	6% of loss amount	\$600,000	\$300,000

When there is a loss of \$2 million on a risk with total insured value of \$3.7 million, Surplus Treaty 2 does not apply. This treaty does not apply because the risk does not exceed \$5 million. Only the Quota Share Treaty and Surplus Treaty 1 apply. The proportional treaties share the loss proportionally as shown in the last two columns of the following table:

Treaty	\$4 million risk proportional share calculation formula	Proportional share of loss	Actual monies tendered on the \$2 million loss
Surplus 2	N/A since the total risk < \$5 million	0%	\$0.00
Surplus 1	100% x 2.7 million/3.7 million	73%	\$1.46 million
Quota share	(40% x \$1 million)/3.7 million	11%	\$220,000
Carrier's share	(60% x \$1 million)/3.7 million	16%	\$320,000

## Proportional Facultative Agreements

Proportional facultative agreements differ in several ways from proportional treaties.

Proportional treaties define how much risk within the coverage group is ceded to the reinsurer in terms of either:

- A percentage share—the *quota share*
- Layers to be ceded—the *surplus*

A treaty applies to all risks within the scope of the treaty. New risks within the coverage group signed by carrier are automatically covered by existing treaty.

Facultative agreements, on the other hand, reinsurance a specific risk. The agreement can simply cede a monetary value, such as \$2 million of the risk, or a percentage, such as 15% of the risk. If the agreement cedes a monetary value, the system determines a percentage share for determining ceded loss. In practice, the agreement might represent the layer above the highest surplus treaty.

A proportional facultative agreement, like a proportional treaty, shares premiums and losses from the first dollar.

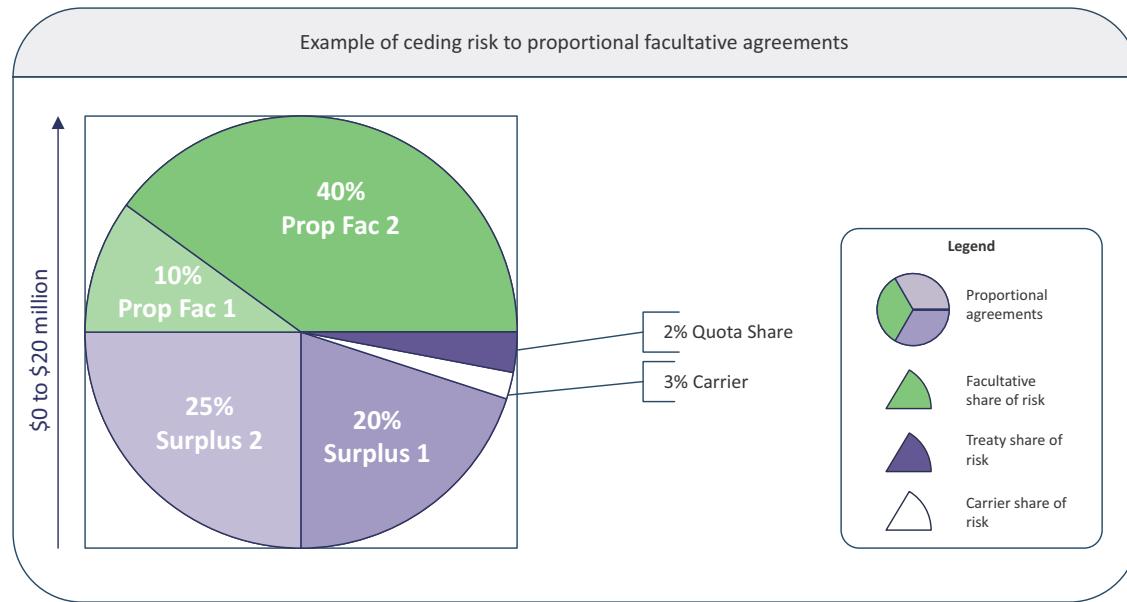
### Example of Ceding Amount of Risk to Proportional Facultative Agreements

This example shows how risk is ceded in a proportional facultative agreement with a monetary amount entered in the **Amount of Risk Ceded** field on the **Facultative** screen. This example builds upon “Example of Ceding Risk to Proportional Treaties” on page 596, which provided reinsurance from \$0 to 10 million on a specific risk. However, in this example the risk is valued at \$20 million. The carrier negotiates two proportional facultative agreements to provide coverage for claims up to \$20 million. One facultative agreement cedes \$2 million in risk. A second facultative agreement cedes \$8 million of risk.

In a reinsurance program, quota share and surplus treaties often provide reinsurance coverage as in the following example. In this example, the risk is equal to the total risk covered by the treaties and facultative agreements put together. If the risk had been smaller, some of the treaties might drop out of the coverage. Also, if the risk were smaller, the smaller risk would replace the \$20 million in the proportional share of risk calculation in the last column.

Treaty	Layers of reinsurance	Amount of risk ceded	Proportional share of risk
<b>Proportional Facultative Agreements</b>			
Proportional Facultative 2		\$8 million	\$8 million of \$20 million = 40%
Proportional Facultative 1		\$2 million	\$2 million of \$20 million = 10%
<b>Proportional treaties</b>			
Surplus 2	From \$5 million to \$10 million	\$5 million	\$5 million of \$20 million = 25%
Surplus 1	From \$1 million to \$5 million	\$4 million	\$4 million of \$20 million = 20%
Quota share	From \$0 to \$1 million ceding 40% of the risk to the reinsurer	\$400,000	\$400,000 of \$20 million = 2%
Carrier's share	From \$0 to \$1 million 60% of the risk retained by the carrier	\$600,000	\$600,000 of \$20 million = 3%

The following illustration shows the coverage provided by the reinsurance program:



When there is a loss of \$20 million or less, the proportional agreements share the loss proportionally, as shown in the last two columns of the following table. In this example, the risk equals the risk limit of the combined treaties:

Agreement	Proportional share of loss	\$20 million loss	\$5 million loss
<b>Proportional facultative agreements</b>			
Proportional facultative 2	40% of loss amount	\$8 million	\$2 million
Proportional facultative 1	10% of loss amount	\$2 million	\$500,000
<b>Proportional treaties</b>			
Surplus 2	25% of loss amount	\$5 million	\$1.25 million
Surplus 1	20% of loss amount	\$4 million	\$1 million
Quota share	2% of loss amount	\$400,000	\$100,000
Carrier's share	3% of loss amount	\$600,000	\$150,000

#### Example of Ceding a Share Percentage to Proportional Facultative Agreements

The previous example shows how risk is ceded to proportional facultative agreements by entering a monetary amount in the **Amount of Risk Ceded** field on the **Facultative** screen. Instead of entering a monetary amount, you can specify a **Ceded Share (%)** field on the **Facultative** screen. For example, the ceding is the same if the two proportional facultative agreements specify 40% and 10% instead of \$8 million and \$2 million, respectively.

## Non-proportional Agreements

Reinsurance Management provides non-proportional reinsurance for both treaties and facultative agreements.

In non-proportional reinsurance there is no proportional ceding of the risk and no proportional sharing of the premium or the losses. The carrier pays the entire loss up to an agreed amount called the *attachment point*. The reinsurer then pays all or part of the loss that exceeds the attachment point up to a limit previously agreed upon by the carrier and reinsurer. The reinsurance premium charged by the reinsurer does not have a direct proportional relationship to the amount of loss that the reinsurer is responsible for.

## Non-proportional Treaties

Reinsurance Management provides the following types of non-proportional treaties:

- **Excess of Loss** – The reinsurer pays a percentage of the amount of a loss in excess of a specified retention for each risk coverage. An excess of loss treaty has an attachment point and coverage limit, and coverage applies to one risk.
 

For example, if a storm destroys 10 covered locations, the limit is applied 10 times, once for each location.
- **Net Excess of Loss** – Similar to an excess of loss agreement. However, *net excess of loss* covers losses net of any recoveries from excess of loss or proportional agreements. A net excess of loss treaty has an attachment point and coverage limit.
- **Per Event** – Cover aggregate losses from an event with multiple risks. A per event agreement is similar to a net excess of loss agreement. The carrier determines its net loss after deducting any amounts recoverable from per risk proportional or non-proportional agreements. Then the per event agreement provides coverage if those net losses are above the attachment point of the per event agreement.
 

Per event treaties are typically catastrophe, for property, or clash cover, for liability.
- **Annual Aggregate** – Similar to a per event treaty. An annual aggregate treaty provides aggregate coverage, net of any per risk coverage or more specific aggregate coverage, such as per event coverage. The annual aggregate treaty covers total losses for an entire book of business for a defined period of time. The period of time is usually one program year. Annual aggregate treaties are defined to start at a specified attachment point or for losses above a specified loss ratio. In either case, the treaty defines a coverage limit. The coverage limit is the maximum amount the reinsurer pays under the treaty, not the top of a layer as in other non-proportional treaties.
 

For example, an aggregate agreement provides reinsurance for net losses to all covered buildings after recovering per risk reinsurance for each building.

See “Example of Ceding Risk to Per Event and Annual Aggregate Treaties” on page 602.

### Example of Ceding Risk to a Single Excess of Loss Treaty

An excess of loss treaty has an attachment point of \$1 million, and a coverage limit of \$3 million with 0% carrier share. The reinsurer does not cover the first \$1 million of any loss, but does cover 100% of the loss above \$1 million up to the limit of \$3 million. The reinsurer provides \$2 million in excess coverage, the Coverage Limit minus the Attachment Point, often referred to as *\$2 million in excess of \$1 million*.

Treaties	Layers of reinsurance
	From \$3 million and up, the carrier provides 100% coverage.
Excess of Loss	Attachment point \$1 million Coverage limit \$3 million
	From \$0 to \$1 million, the carrier provides 100% coverage.

Losses would be covered by this agreement as follows:

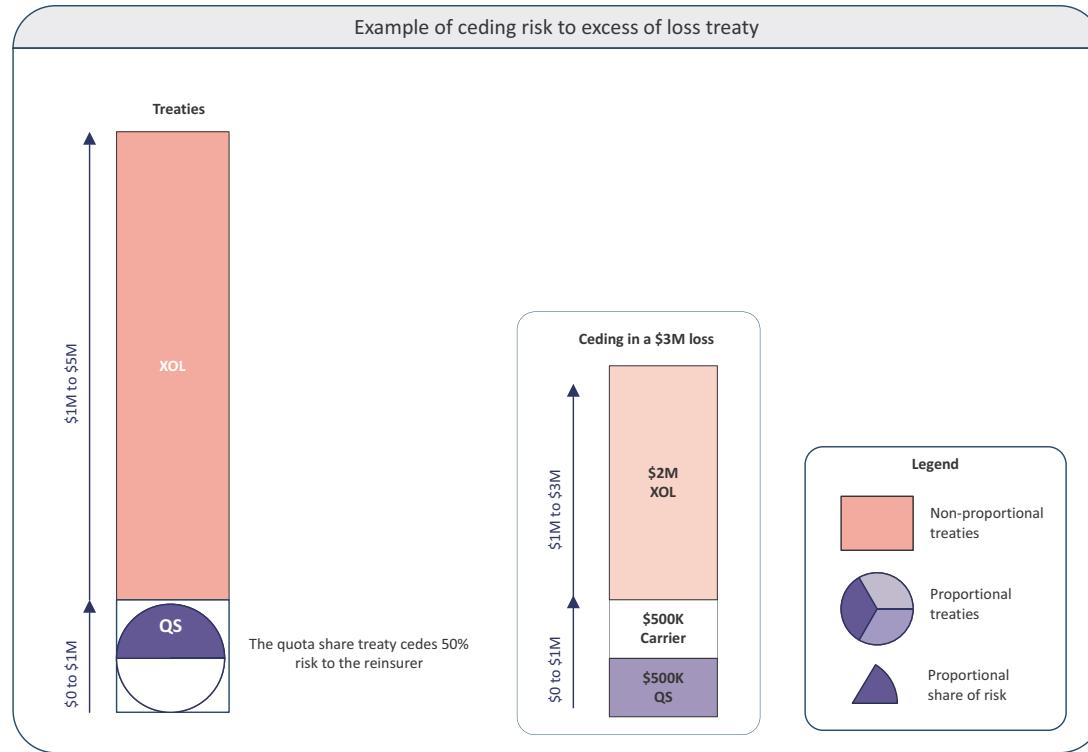
- \$900,000 loss – The reinsurer pays nothing because it is under the \$1 million attachment point.
- \$2,500,000 loss – The carrier pays the first \$1 million, and the reinsurer pays the next \$1,500,000.
- \$4,500,000 loss – The carrier pays the first \$1 million. The reinsurer pays the next \$2 million up to the reinsurance limit of \$3 million. The carrier pays the last \$1.5 million, unless the carrier has another reinsurance agreement that covers a higher band of losses, which would typically be the case.

### Example of Ceding Risk to Multiple Excess of Loss Treaties

The carrier has a program that contains two treaties. A quota share treaty covers 50% up to \$1 million. An excess of loss treaty covers \$4 million in excess of \$1 million.

Treaty	Layers of reinsurance
Excess of loss	Attachment point: \$1 million Coverage limit: \$5 million
Quota share treaty	50% up to \$1 million

If there is a \$3 million loss, the carrier pays a 50% share of the first \$1 million. The excess of loss agreement pays the \$2 million above the \$1 million attachment point. The carrier's gross retention is \$1 million, where the excess of loss attaches, and total net retention for any loss under \$5 million is \$500,000.



### Example of Ceding Risk to a Net Excess of Loss Treaty

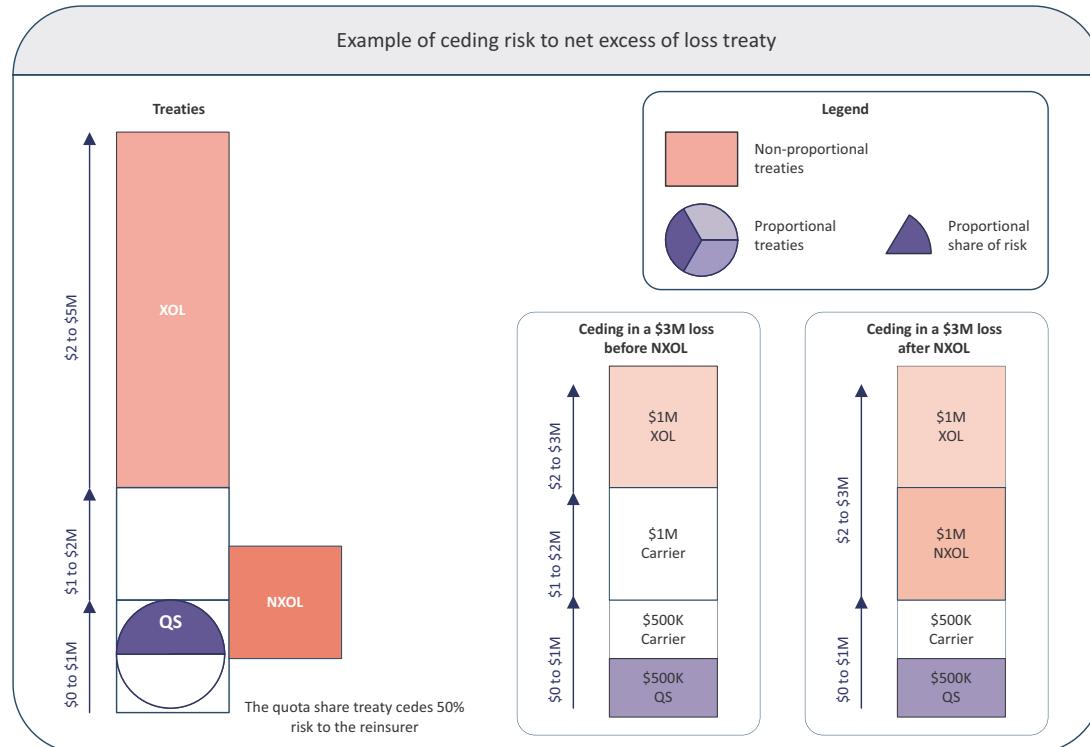
The carrier has a program that contains three treaties:

- A quota share treaty covers 50% up to \$1 million.
- An excess of loss treaty covers \$3 million in excess of \$2 million.
- A net excess of loss treaty covers \$1 million in excess of \$500,000.

Layers of reinsurance	
Treaties	
Excess of loss	Attachment point: \$2 million Coverage limit: \$5 million
Quota share treaty	Attachment point: \$500,000 Coverage limit: \$1 million

Layers of reinsurance	
Net treaties	
Net excess of loss	50% up to \$1 million

If there is a \$3 million loss, the carrier pays a 50% share of the first \$1 million and 100% of the next \$1 million. The excess of loss pays the \$1 million above \$2 million. The carrier's net loss is \$1.5 million, but the carrier collects \$1 million from the net excess of loss agreement for the amount of net loss above \$500,000. The carrier's *gross retention* is \$2 million, where the excess of loss attaches, and total net retention for any loss under \$5 million is \$500,000.



### Example of Ceding Risk to Per Event and Annual Aggregate Treaties

A carrier might be willing to hold a \$1 million net retention for any one risk to property. However, if there are widespread losses from a single catastrophic event such as a tornado or flood, 100 separate losses could add up to \$100 million in retained risk. To protect against a loss of this magnitude, the carrier can have a per event agreement to provide coverage for \$100 million in excess of \$20 million. The carrier retains \$20 million in aggregate net risk. The carrier collects \$80 million from the per event agreement in the case of a \$100 million net loss (after per risk insurance) from a single event.

Annual aggregate treaties provide reinsurance coverage for multiple catastrophic events in a single year. For example, the carrier had planned for 2,000 slip and fall liability losses in a year, but there are claims for 10,000. The carrier's risk retention is unacceptably high. To protect against this eventuality, the carrier negotiates an annual aggregate treaty to cover yearly net losses for \$500 million in excess of \$200 million. The annual aggregate treaty cedes 75% of the risk to the reinsurer.

### Non-proportional Facultative Agreements

Non-proportional facultative agreements can be excess of loss or net excess of loss agreements.

### Excess of Loss

Non-proportional facultative agreements are usually excess of loss agreements.

If a facultative excess of loss agreement insures amounts above other excess of loss agreements, it provides another layer of coverage when no standard treaty is in place. There is no difference from a standard excess of loss situation.

However, if a facultative excess of loss agreement insures amounts above a set of proportional agreements, the behavior is different. When a set of proportional treaties are in place, the idea is to share risks up to the limit of the highest surplus, such as \$2 million. For larger risks, a facultative excess of loss agreement can remove the potential for losses larger than \$2 million. The risk still looks like a \$2 million risk to all the proportional participants.

The carrier charges a premium to cover the cost of the facultative excess of loss agreement plus other costs such as commissions to agents. Since all proportional participants benefit from the facultative excess of loss agreement, the premium is shared proportionally after deducting the cost of the facultative excess of loss agreement.

### Net Excess of Loss

The other type of non-proportional facultative agreement is a net excess of loss agreement. This agreement provides reinsurance after proportional reinsurance and protects only the carrier's share of the risk.

The net excess of loss premium is not deducted in advance of determining what is shared among the proportional participants.

## Summary of Agreement Types

Reinsurance agreements are categorized into different types based on how the risk is shared. The agreement records the parameters to be used in the calculation of how to divide the risk and how to distribute the premiums.

The following table shows the types of agreements in the default configuration. The marked cells indicate that the item applies to that agreement type.

Agreement Type	Treaty	Facultative	Per Risk	Aggregate	Policy Attachment	Loss Date Attachment
<b>Non-proportional</b>						
Annual Aggregate	●			●		●
Per Event	●			●		●
Excess of Loss	●		●		●	●
Net Excess of Loss	●		●		●	●
Facultative Excess of Loss		●	●		●	
Facultative Net Excess of Loss		●	●		●	
<b>Proportional</b>						
Quota Share	●		●		●	
Surplus	●		●		●	
Facultative Proportional	●		●		●	

The Policy Attachment column shows the types of agreements that apply to all losses against the policy for the entire term.

The Loss Date Attachment column shows the types of agreements that apply to a policy on the loss date. PolicyCenter chooses the agreement that is in effect on the loss date rather than the start date of the policy.

Excess of loss and net excess of loss treaties can be specified as either policy attachment or loss date attachment. You set these values in the **Loss Attachment Basis** field on the **Treaty** screen.

**See also**

- “How PolicyCenter Attaches Agreements to Policies” on page 604

## How PolicyCenter Links Reinsurance to Policies

PolicyCenter applies one or more reinsurance programs to a policy. PolicyCenter links the policy to each agreement in the program.

### How PolicyCenter Attaches Programs to Policies

In a submission, PolicyCenter selects reinsurance programs that:

- Are in effect while the policy is in effect.
- Provide coverage for each reinsurance coverage group of coverages on the policy.
- Have the same currency as the TIV/SI of the risk. This applies to multicurrency policies.

For each risk on the policy, PolicyCenter tries to find one program for each reinsurance coverage group. For example, if a location has two coverage groups, PolicyCenter will link that location to a reinsurance program for each coverage group.

**See also**

- “Multicurrency and Reinsurance” on page 521

### Invalid Draft Reinsurance Programs Can Cause Problems

In PolicyCenter, you can create and save draft reinsurance programs that do not pass validation. Invalid programs can include things such as inconsistent dates, inappropriate retention levels, inconsistent reinsurance coverage groups, and inconsistent currencies. All active programs are valid.

There is no problem if these invalid draft reinsurance programs are not attached to policies. However, in the default configuration, PolicyCenter may attach a draft program to a policy if it cannot find an appropriate active program. PolicyCenter evaluates the reinsurance using the draft program, which may not be valid and may not meet other business requirements. Additionally, the currencies in this program may be inconsistent with each other or with the total insured value or sum insured of the risk. If this is the case, the reinsurance calculation may fail ungracefully.

You can avoid these problems by ensuring that reinsurance programs and their agreements that might attach to policies pass validation and meet applicable business and regulatory requirements.

### How PolicyCenter Attaches Agreements to Policies

PolicyCenter attaches reinsurance agreements to policies to provide reinsurance coverage. The date that the reinsurance agreement attaches, and how long the agreement provides coverage is based on whether the agreement is a *policy attachment* or *loss date attachment* agreement.

- **Policy attachment agreements** – *Policy attachment* treaties apply to all losses against the policy for the entire term.

All proportional agreements and all facultative agreements attach to the policy.

- **Loss date attachment agreements** – These treaties apply to the policy at the loss date, rather than the policy effective date. In these agreements, PolicyCenter chooses the agreement that is in effect on the loss date rather than the start date of the policy.

In the following table, the marked cells indicate the agreement type.

Agreement	Policy Attachment	Loss Date Attachment
<b>Non-proportional treaties</b>		
Annual Aggregate Treaty		●
Per Event Treaty		●
Excess of Loss Treaty	●	●
Net Excess of Loss Treaty	●	●
<b>Non-proportional facultative agreements</b>		
Facultative Excess of Loss	●	
Facultative Net Excess of Loss	●	
<b>Proportional treaties</b>		
Quota Share	●	
Surplus	●	
<b>Proportional facultative agreements</b>		
Facultative Proportional	●	

Excess of loss and net excess of loss treaties can be specified as either policy attachment or loss date attachment. You set this in the **Loss Attachment Basis** field on the **Treaty** screen.

In a multicurrency system, PolicyCenter attaches a reinsurance agreement that has the same currency as TIV/SI of the risk.

#### See also

- “Multicurrency and Reinsurance” on page 521

### How PolicyCenter Selects a Projected Program

At a particular point in time, the programs for the coming year may not have been entered into PolicyCenter, or may not be finalized. Regardless, PolicyCenter needs to determine if reinsurance coverage will be adequate. For example, you are processing renewals in 10/2011 for policies effective in 1/2012, but the 2012 programs have not yet finalized. PolicyCenter estimates the reinsurance coverage if an active program is not yet available. This behavior avoids having policies appear to be inadequately reinsured, if the next year’s programs are not yet finalized.

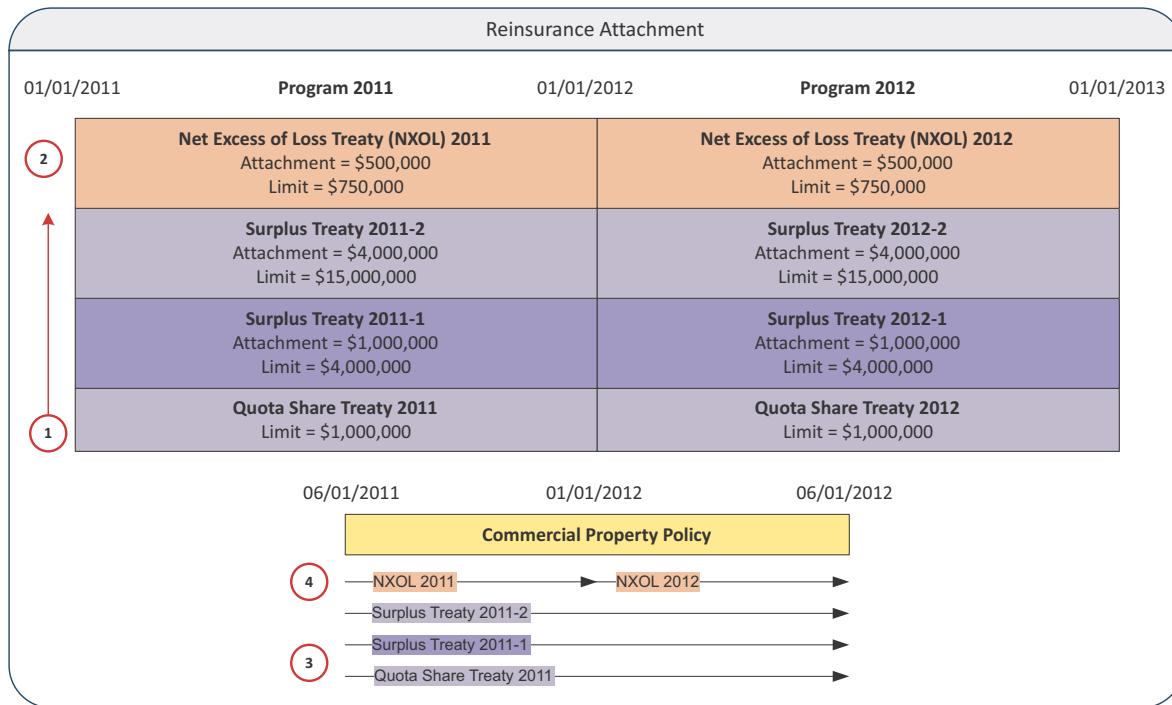
Programs for succeeding years may either be in draft status or not yet entered into PolicyCenter. There may be no active program for the date range and coverage group. If so, PolicyCenter selects the program for a risk in the following order:

1. Draft program for the date range and coverage group
2. Prior year active program for the coverage group
3. If not matches are found, PolicyCenter writes an error to the log file, but does not block progress on the policy.

If PolicyCenter selects a draft or prior year program, the agreements from that program are marked as **Projected** in the policy. PolicyCenter uses projected programs only for testing adequacy. PolicyCenter does not consider these projected reinsurance coverages when ceding premium or sending reinsurance coverage information to a claim system.

## Policy Attachment Example

In the following illustration, PolicyCenter has two active reinsurance programs for property. Program 2011 starts on 01/01/2011 and is effective for the whole year. Program 2012 continues the same reinsurance coverage for 2012. A commercial property policy starts on 06/01/2011 and ends one year later.



The numbers in the illustration show the following:

1. Both programs have three policy attachment treaties:

- One quota share treaty
- Two surplus treaties

2. Both programs have one loss attachment treaty:

- One net excess of loss treaty

**Note:** In this example, the **Loss Attachment Basis** of the net excess of loss treaty is set to **Loss Date Attachment**. For net excess of loss treaties, this field can also be set to **Policy Attachment**.

3. On the commercial property policy, the policy attachment treaties from Program 2011 attach on the effective date, 06/01/2011. The policy attachment treaties from Program 2011 provide reinsurance coverage for the whole policy term, through 06/01/2012. Although the Program 2011 treaties end on 01/01/2012 in the middle of the policy term, the policy is still protected by the policy attachment treaties in Program 2011.

4. The NXOL loss attachment treaty from Program 2011 attaches to the policy on the effective date, 06/01/2011. This treaty ends on 01/01/2012, and the NXOL treaty from Program 2012 attaches to the policy on this date.

**Note:** A midterm policy change does not change the set of treaties that apply to a risk because only the policy period effective date matters for policy attachment treaties. Only the effective date of the program matters for loss attachment treaties.

### See also

- “Adding Reinsurance to a Policy” on page 616
- “Reinsurance Configuration Plugin” on page 467 in the *Integration Guide*

## How PolicyCenter Calculates Ceded Premiums

In a reinsurance agreement, the reinsurer takes responsibility for part of the risk that the carrier assumed from the insured. Therefore, the carrier cedes part of the premium to the reinsurer.

Ceded premiums are based on the premiums paid by the insured. For a given risk and set of coverages, one or more reinsurance agreements provides reinsurance coverage. The carrier calculates how much of the direct premium is ceded to each agreement.

**IMPORTANT** PolicyCenter does not calculate ceded premiums for non-proportional treaties.

However, you can add this calculation to the reinsurance ceding plugin. For more information about configuring reinsurance ceding, see “Reinsurance Ceding Plugin Implementations” on page 470 in the *Integration Guide*.

### Premium ceding in a proportional treaty

For proportional treaties, the carrier cedes a share percentage of the premium for the total risk. For example, there is a total risk of \$1 million, and a quota share treaty transfers 50% proportional share of the risk up to \$1 million to the reinsurer. Because the quota share treaty assumes 50% of the total risk, the carrier cedes 50% of the total premium to the reinsurer.

If the total risk is above the coverage limit, the ceded premium is adjusted to reflect the percentage of the total risk actually assumed. (There is an exception for excess of loss. See the premium calculation in “Excess of Loss” on page 603.) For example, there is a quota share treaty with a \$1 million limit and a 40% proportional share, but the total risk is \$2 million. Because the quota share treaty only assumes 20% of the total risk, the reinsurer receives 20% of the premium.

The premium calculation is the same if the total risk is above the coverage limit, and surplus treaties provide reinsurance above the coverage limit.

### When PolicyCenter calculates ceded premiums

PolicyCenter calculates ceded premiums if both of the following are true:

- The policy has been issued.
- There is at least one active program that applies to the risks on the policy

**IMPORTANT** In the default configuration, PolicyCenter calculates the ceded premiums and commissions and stores the values in a database table. You can integrate with an accounts payable system that processes the ceded premiums and commissions.

### Commissions

In addition to earning ceded premiums, the reinsurer pays the carrier a commission that is a percentage of the ceded premium. All agreement types have a field for commission percentage. If there is no commission, set the percentage to 0. For example, many non-proportional agreements do not pay a commission.

The reinsurer pays a commission for several reasons, including:

- The reinsurer shares the insurance business without the cost of acquiring the customer. Costs include marketing and sales.
- The reinsurer does not provide the customer with services such as adjudicating claims and billing.

The carrier pays the reinsurer the ceded premium minus the commission. For accounting purposes, the ceded premium and the commission are kept as separate values.

### Multicurrency and Ceding

In a multicurrency system, multiple currencies do not affect premium ceding for proportional agreements. An agreement that cedes a fixed amount may require a currency conversion. For more information, see “Ceding Premium in a Multicurrency Policy” on page 522 in the *Application Guide*.

## Calculating Ceded Premiums

PolicyCenter calculates the ceded premium as follows:

### Step 1: Cede Premium to Facultative Excess of Loss and Facultative Proportional Agreements

If there are facultative excess of loss (Fac XOL) agreements with flat ceding amounts, cede those amounts. Cede any markup on these agreements. Calculate the gross net premium (GNP) as follows:

$$\text{GNP} = \text{Written Premium} - \text{Fac XOL Ceded Premium} - \text{Fac Markup}$$

**Note:** When calculating the GNP in the default configuration, cedings are only deducted from facultative excess of loss agreements and not from excess of loss treaties. If you expect to have programs with excess of loss treaties above proportional layers, you can configure Reinsurance Management to calculate GNP for this case.

The proportional treaties share the gross net premium (GNP) after ceding to any excess of loss agreements.

Net excess of loss and aggregate treaties are not deducted because they apply only to the carrier’s net risk after deducting risk ceded to proportional agreements.

### Step 2: Cede Premium to Proportional Agreements

When you calculate the ceded premiums for proportional risks, you calculate the actual percentage of risk that is held by each of the proportional agreements. A modified TIV is the denominator.

Modified Total Insured Value (TIV) represents the amount of risk shared proportionally between the carrier the proportional agreements. The modified TIV is the TIV minus any amounts that an excess of loss treaty or facultative agreement covers. Net excess of loss and aggregate treaties are not deducted because they apply only to the carrier’s net risk after deducting amounts ceded proportionally.

1. Calculate the modified Total Insured Value (TIV):

$$\text{Modified TIV} = \text{Actual TIV} - \text{Amount of Risk in XOL Layer}$$

2. For each proportional agreement, calculate the ceded premium:

- a. Calculate the proportional share of risk

If this is a facultative proportional agreement that specifies a Ceded Share (%), use this formula:

$$\text{Proportional Share of Risk} = \text{Ceded Share Percentage}$$

Otherwise, use the following formula:

$$\text{Proportional Share of Risk} = \text{Amount of Risk Ceded} / \text{Modified TIV}$$

- b. Calculate the ceded premium:

If this is a facultative proportional agreement that specifies a flat amount for ceded premium, then cede that amount.

Otherwise, use the following formula:

$$\text{Prop Ceded Premium} = \text{GNP} * \text{Proportional Share of Risk}$$

### Step 3: Cede Premium to Facultative Net Excess of Loss Agreements

After ceding premium to proportional agreements, PolicyCenter cedes the premium to the facultative net excess of loss agreements. The ceded premium is always a flat amount.

**See also**

- “Viewing Ceded Premiums” on page 618
- “Reinsurance Ceding Plugin” on page 470 in the *Integration Guide*

## Ceded Premiums Example

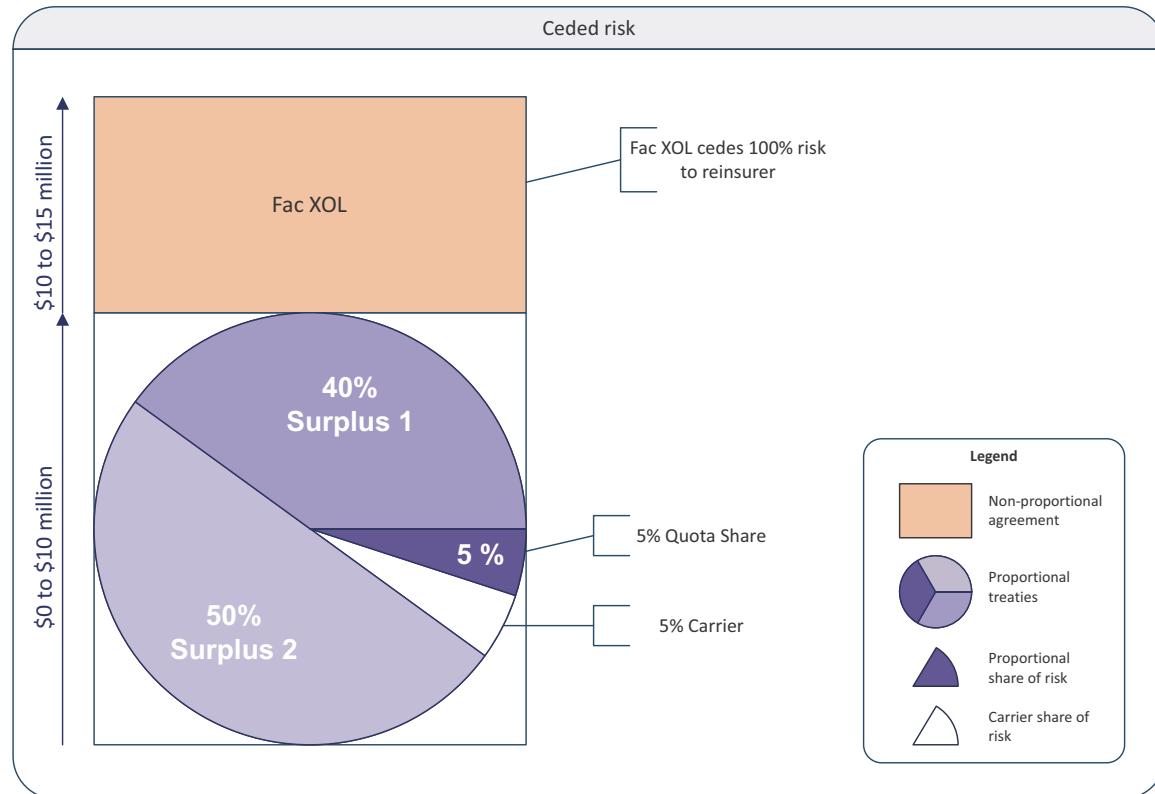
This example shows how PolicyCenter cedes premiums to several proportional agreements and to the facultative excess of loss agreement above the proportional agreements.

The following example is for a building with a \$15 million TIV.

The reinsurance program attached to this building contains the following agreements:

Agreement	Layers of reinsurance	Amount of risk ceded	Proportional share of risk
Facultative excess of loss	\$10 million to \$15 million	\$5 million	Not applicable
<b>Proportional Treaties</b>			
Surplus treaty 2	\$5 million to \$10 million	\$5 million	\$5 million of \$10 million = 50%
Surplus treaty 1	\$1 million to \$5 million	\$4 million	\$4 million of \$10 million = 40%
Quota share treaty	\$0 to \$1 million ceding 50% of the risk to the reinsurer	\$500,000	\$500,000 of \$10 million = 5%
Carrier's share	\$0 to \$1 million, carrier retaining 50% of the risk	\$500,000	\$500,000 of \$10 million = 5%

The program provides reinsurance coverage as shown in the following illustration.



The Facultative Excess of Loss agreement takes \$5 million of this risk, so the modified TIV is \$10 million.

The total written premium is \$11,250. The ceded premium for the Facultative Excess of Loss agreement is a flat amount of \$1000 with a 25% markup (\$250). Therefore, the gross net premium (GNP) is \$10,000.

The pie chart shows the proportional share of risk for each proportional treaty.

The ceded premium for the quota share treaty is calculated as follows:

1. Calculate the proportional share of risk:

$$\text{Proportional Share of Risk} = \text{Amount of Risk in Layer} / \text{Modified TIV}$$

$$10\% = \$1 \text{ million} / \$10 \text{ million}$$

2. Calculate the proportional ceded premium:

$$\text{Prop Ceded Premium} = \text{GNP} * \text{Percentage of Premium}$$

$$\$500 = \$10,000 * 5\%$$

The premium ceded to the proportional treaties are:

Proportional treaty	Proportional share of risk	Proportional ceded premium
Surplus treaty 2	50%	\$5,000
Surplus treaty 1	40%	\$4,000
Quota share treaty	5%	\$500

## Shared Reinsurance Agreements

It is common for multiple reinsurance companies to participate in a reinsurance agreement. Each reinsurer participates for a percentage of the agreement, and the percentages add up to 100%. You specify a shared agreement by adding more than one reinsurer to the **Agreement Participants** tab of an agreement.

Shared reinsurance agreements are much the same as reinsurance agreements with a single reinsurer. The carrier determines the ceded premiums, commissions, and loss recoverables for the agreement. The carrier divides the net amount due to the participating reinsurers according to their participating percentage. PolicyCenter tracks amounts against the agreement as a whole, but keeps track of the schedule of participants. Therefore, an external accounts payable system can divide these amounts among the participants.

### See also

- “Agreement Participants Tab” on page 626

## Differential Rates

The rate of commission charged or premium ceded to each reinsurer in an agreement can differ. By using differential rates, you can set different values for each agreement participant.

For example, two reinsurers share in the premiums and losses at 50% each. However, one reinsurer agrees to a 10% commission and the other agrees to 15%. The overall commission rate for the agreement is:

$$(50\% * 10\%) + (50\% * 15\%) = 12.5\%$$

PolicyCenter determines the overall commission for the agreement based on this blended commission rate, but the amount of commission deducted from the premiums payable to each reinsurer differs.

You set differential commission rates by selecting **Set Differential Commission Rates** on the **Facultative or Treaty** screen. After you select this, you can enter a **Commission %** for each agreement participant.

For facultative agreements that cede a flat amount, you can set differential rates for ceded premium by selecting **Set Differential Flat Premium** on the **Facultative** screen. After you select this, you can enter a **Flat Premium** for each agreement participant.

For treaties that cede a percentage, you can set differential ceding rates by selecting **Set Differential Ceding Rates** on the **Treaty** screen. After you select this, you can enter a **Ceding Rate (%)** for each treaty participant.

**See also**

- “Fields” on page 622

## Location Groups

Location groups provide a way to group locations that are located near each other. A carrier can use location groups to help assess their total risk for a given location.

A carrier can provide coverage on different policies that are considered a single risk for reinsurance. These coverages can include:

- Multiple stores within the same shopping complex
- Multiple offices within a large office building
- Multiple apartments or condominiums within same building

For example, a carrier has commercial property policies for two different accounts. The properties in these account are located close enough to each other that a single fire could affect both locations. When assessing the total potential risk, the carrier would like to know about the close proximity of these two, and possibly other, locations within the range of a single event.

In PolicyCenter, you can search for all risks of a designated type that are within a specified distance from a central location. You can combine the nearby risks into a location group.

**Note:** Searching for nearby locations requires that geocoding is enabled.

In the default configuration, location groups simply group locations. You can use and configure location groups for a variety of purposes including:

- For generating reports about a location group.
- For underwriting rules that check whether the sum of total insured value of all risks in a location group exceeds a threshold. PolicyCenter raises an underwriting issue if the value exceeds the threshold.

**See also**

- “Creating a Location Group” on page 618
- “Geocoding Locations” on page 344
- “Configuring Geocoding” on page 354



# Reinsurance Management User Interface

This topic describes the Reinsurance Management user interface in PolicyCenter.

This topic includes:

- “Working with the Reinsurance Tab” on page 613
- “Working with Reinsurance Management in Policies” on page 616
- “Reinsurance Management Screens” on page 621

## Working with the Reinsurance Tab

You can view and define reinsurance agreements and programs on the Reinsurance tab. This topic describes how to work with Reinsurance Management from the Reinsurance tab.

- “Searching for Agreements” on page 613
- “Creating a New Treaty” on page 614
- “Creating a New Program” on page 614
- “Editing a Program” on page 615
- “Creating a New Facultative Agreement” on page 615
- “Validating an Agreement” on page 616
- “Making an Agreement Active” on page 616

### Searching for Agreements

You can search for all agreements, whether treaties or facultative agreements, on the **Reinsurance → Search Agreements** screen. You can use search criteria to narrow the search.

### To search for all agreements

1. Click **Agreements** in the left sidebar to view the **Search Agreements** screen.
2. Click **Reset**.
3. In **Effective Date**, select **<none selected>**.  
PolicyCenter displays all agreements.

### To narrow the search for agreements

1. Click **Agreements** in the left sidebar to view the **Search Agreements** screen.
2. Narrow the search for agreements by selecting any of the available agreement search parameters.
  - Specify **Agreement Number** or **Name** to return agreements that start with the specified string.
  - Specify **Coverage Group** to limit the search to agreements that contain the specified reinsurance coverage group.
  - Specify **Status**, **Type**, and **Arrangement** fields to return results that match the corresponding field exactly. The fields **Type** and **Arrangement** can have logically conflicting settings which result in matching zero agreements. For example, if you set **Type** to **Surplus Treaty** and **Arrangement** to **Facultative**, the search always returns zero matches.
  - Specify the **Effective Date** field to return agreements with a start date within the search date range.
  - Specify **Currency** to filter the agreements by currency.

## Creating a New Treaty

1. Click the **Reinsurance** tab.
2. Select **Actions** → **New Treaty** and select one of the following treaty types:
  - **Per-Event Treaty**
  - **Annual Aggregate Treaty**
  - **Quota Share Treaty**
  - **Surplus Treaty**
  - **Excess of Loss Treaty**
  - **Net Excess of Loss Treaty**
3. Enter values for all required fields.
4. On the **Agreement Participants** tab, add at least one participant. A participant is a reinsurer participating in this agreement. For more information about this tab, see “**Agreement Participants Tab**” on page 626.
5. On the **Applies To** tab, add one or more **Included Coverage Groups** such as property or auto liability. For more information about this tab, see “**Applies To Tab for Agreements**” on page 627.
6. Click **Update** to create the treaty.

## Creating a New Program

A reinsurance program is a set of treaties put together by the carrier. In a multicurrency system, the program and its treaties must have the same currency.

1. Click the **Reinsurance** tab.
2. Select **Actions** → **New Program**.
3. At a minimum, enter a **Name**, **Effective Date**, and **Expiration Date**.

4. On the **Treaties** tab, click **Add** to search for **Per Risk** and **Aggregate** agreements. For more information about this tab, see “Treaties Tab for Programs” on page 628.
5. On the **Applies To** tab, add one or more **Included Coverage Groups** such as property or auto liability. For more information about this tab, see “Applies To Tab for Programs” on page 629.
6. Click **Update** to create the program.

**See also**

- “Reinsurance Program Basics” on page 590
- “Reinsurance Program Screen” on page 628

## Editing a Program

On an existing program, click **Edit** to make the following types of changes:

- Change the **Name**
- Change the **Expiration Date**
- Change the **Target Net Retention**
- Change the **Single Risk Maximum**
- Add or remove **Per Risk** or **Aggregate** treaties
- Add or remove **Included Coverage Groups**

If you edit a program that is in effect, PolicyCenter recalculates ceded premiums on all policies to which this program has been applied. A program is in effect from the **Effective Date** to the **Expiration Date**.

**See also**

- “Reinsurance Program Screen” on page 628

## Deleting a Program

On an existing program, you can click **Delete** to remove a program.

### Disabling a Program that Has Attached Policies

If a program is attached to at least one policy, the **Delete** button is not accessible. If you do not want this program to attach to any more policies, **Edit** the program, and set the **Effective Date** and **Expiration Date** to the same value.

## Creating a New Facultative Agreement

1. Click the **Reinsurance** tab.
2. Select **Actions** → **New Facultative Agreement** and select one of the following agreement types:
  - **Proportional Facultative Agreement**
  - **Excess of Loss Facultative Agreement**
  - **Net Excess of Loss Facultative Agreement**
3. Enter values for all required fields.
4. On the **Agreement Participants** tab, add at least one participant. A participant is a reinsurer participating in this agreement. For more information about this tab, see “Agreement Participants Tab” on page 626.
5. Click **Update** to create the agreement.

## Validating an Agreement

You can run validation on an agreement that is in read only mode. Validation displays verification errors or verification that the agreement is valid. Validation runs automatically if you try to make the agreement active.

Validation checks include, but are not limited to, the following:

- Required fields are filled in and related fields are not logically conflicting.
  - The **Expiration Date** must be greater than the **Effective date**.
  - The coverage limit must be greater than or equal to the attachment point. In special cases the coverage limit must be equal to the attachment point.
- The ceding rate must be greater than or equal to 0.

### To validate an agreement

1. Select **Reinsurance** → **Agreements**.
2. Enter search criteria, and click **Search**.
3. In the **Agreement Number** column of **Search Results**, click the link to an agreement.  
The agreement must not be being edited.
4. Click **Validate**.

## Making an Agreement Active

An agreement can have a status of **Active** or **Draft**. Click the **Reinsurance** tab. Make the agreement active when it is finalized and ready for use. You can make an agreement active in one of the following ways:

- On the **Treaty or Facultative** screen that is in read-only mode, click the **Make Active** button.
- On the **Search Agreements** screen, select one or more agreements in the **Search Results** tab and click **Make Active**.
- A set of treaties in a program can be made active when the program is made active.

## Working with Reinsurance Management in Policies

Reinsurance agreements are applied to policies. This topic describes how to work with the **Reinsurance** screen in a policy or policy transaction.

- “Adding Reinsurance to a Policy” on page 616
- “Creating a Location Group” on page 618
- “Viewing Ceded Premiums” on page 618
- “Modifying the Gross Retention” on page 619
- “Adding or Linking to a Facultative Agreement” on page 620
- “Editing Ceding Parameters” on page 621

## Adding Reinsurance to a Policy

Reinsurance is added to a policy at quote. In these instructions, you create a commercial property policy on the Wright Construction account by using the large sample data set. For more information, see “Installing Sample Data” on page 52 in the *Installation Guide*.

When you install the sample data, PolicyCenter creates programs and one-year reinsurance agreements that are effective on the day you install the data. These instructions have you set the policy effective date one month in the future to see what happens when a policy is in effect beyond the end of the agreement.

1. As the underwriter, aapplegate, go to the Wright Construction account.

2. Select Actions → New Submission.
3. Set the Default Effective Date to one month in the future.
4. Select the Commercial Property product.
5. Advance to the Building and Locations screen.
  - a. In the Actions column for the primary location, select Add Building → New Building.
  - b. In Property Class Code, enter 0010.
  - c. For Coverage Form, select Building and Personal Property.
  - d. Click the Coverages tab.
  - e. In Business Income Coverage → Income Limit - Not Mfg or Rental, enter 1,000,000.
  - f. Click OK.

6. Quote the policy.

Under Tools in the left sidebar, the Reinsurance menu item appears.

The Reinsurance plugin assigns reinsurance programs to the policy. For more information see, “Reinsurance Plugin” on page 461 in the *Integration Guide*.

7. Select Tools → Reinsurance.

PolicyCenter displays the Reinsurance screen.

On the Coverage tab, PolicyCenter displays a list of Reinsurable Risks. PolicyCenter displays an Applicable Reinsurance section if it was able to assign reinsurance.

8. Select a risk to display Risk Details.

In this example, the View As Of drop-down list displays two date ranges. The first date range begins on the policy effective date and extends until the first program ends. The second date range begins when the first program ends and extends until the end of the policy term. PolicyCenter attaches agreements to the policy based on whether the agreement is specified as policy attachment or loss date attachment.

9. Select the first date range from the View As Of drop-down list.

The Per Risk tab displays the reinsurance agreements assigned to the currently selected risk. In this example, the Reinsurance plugin selected the Property PY2011 program. The per risk treaties in this program appear on the Per Risk tab in the submission. The aggregate treaties in this program appear on the Aggregate tab.

For more information, see “How PolicyCenter Links Reinsurance to Policies” on page 604.

10. Select the second date range from the View As Of drop-down list.

The quota share and surplus treaties are policy attachment treaties which attach when the policy starts and remain for the policy term. Therefore, these treaties are unchanged in the second date range. The Property NXOL PY2011 treaty is a loss date attachment treaty which attaches to the policy for the duration of the treaty.

For the second date range, there is no active program for those dates which contains loss date attachment treaties. Therefore, PolicyCenter looks for draft programs for that date range, but finds none. Then PolicyCenter looks for and finds an active program from the prior year. This active program contains a loss date attachment treaty. PolicyCenter displays the treaty from the first date range and marks it as Projected by using the assumption that the treaty will be extended to another year. If PolicyCenter had found a treaty in draft status, then PolicyCenter would attach the treaty to the policy and mark it as Projected.

For descriptions of the Reinsurance screen fields, see “Reinsurance Screen in the Policy File” on page 630.

**Note:** You can quote, bind, and issue a policy with projected reinsurance agreements.

From the Reinsurance screen in a policy file, you can take additional actions as described in the following topics:

- “Creating a Location Group” on page 618
- “Viewing Ceded Premiums” on page 618

- “Modifying the Gross Retention” on page 619
- “Adding or Linking to a Facultative Agreement” on page 620
- “Editing Ceding Parameters” on page 621

## Creating a Location Group

A location group is useful for assessing reinsurance risk. In PolicyCenter, you can search for nearby locations and assemble them into a location group. Searching for nearby locations requires that you enable geocoding.

### To create a location group

1. Navigate to the Reinsurance screen for a policy.

The Reinsurance screen must contain at least one location in Reinsurable Risks.

2. Under Reinsurable Risks, select a location.

3. Click Search for Nearby Locations in the Risk Details.

PolicyCenter displays the Search for Nearby Locations screen.

On the left, the Search near section of the screen displays the selected location, including the Location Group, if specified.

On the right, the Search criteria has the following fields:

Field	Description
Radius	Enter the radius of the search. <b>Note:</b> Selecting a too large search radius negatively impacts performance.
Units	Required. Select the units, such as miles or kilometers.
Location Group	Enter a location group name.
Coverage Effective	Required. Select an effective date for coverage. The search looks for policies in effect on this date.
Coverage Group	Required. Select a reinsurance coverage group.
Line of Business	Required. Select one or more lines of business.

4. Select search criteria, and click Search.

By default, results appear in order of increasing distance from the selected location.

### See also

- “Location Groups” on page 611
- “Geocoding Locations” on page 344
- “Configuring Geocoding” on page 354

## Viewing Ceded Premiums

In the policy file, the Risk Details tab of the Reinsurance screen has a View Ceded Premiums button.

Ceded premiums are calculated if both of the following are true:

- The policy has been issued.
- All reinsurance agreements on the policy are active.

### To view ceded premiums on a policy

1. Navigate to a policy that has been issued and for which all reinsurance agreements are active.

2. Click **View Ceded Premiums** to view the ceded premiums for the selected risk.
3. Click a link to view ceded premiums in the following ways:
  - **All Transactions**
  - **All agreements for a cost** – View each agreement's ceding in a calculation for each cost
  - **All costs for an agreement** – View the each cost component for an agreement's ceding
  - **An agreement's cedings across time** – View cost ceding across time for each agreement

**Note:** In the default configuration, PolicyCenter calculates the ceded premiums and commissions and stores the values in a database table. You can integrate with an accounts payable system that processes the ceded premiums and commissions.

#### See also

- “How PolicyCenter Calculates Ceded Premiums” on page 607
- “Reinsurance Ceding Plugin” on page 470 in the *Integration Guide*

## Modifying the Gross Retention

The gross retention is the amount of risk retained by the carrier prior to any amount ceded to the first surplus agreement. If there is a quota share agreement in place, then the gross retention defaults to the coverage limit of the quota share treaty. For a specific risk, a carrier may choose to retain less risk than the treaty specifies by lowering the gross retention.

In the policy file, the **Per Risk** tab of the **Reinsurance** screen has an editable **Gross Retention** field. This value defaults to the **Coverage Limit** of the quota share treaty attached to this risk. If this risk does not have a quota share treaty, then the value defaults to the **Max Retention** of the first surplus treaty. The **Coverage Limit** and **Max Retention** are specified on the **Reinsurance → Treaty** screen.

You can modify the gross retention for each risk on a policy. The value can be less than or equal to the default value.

If you modify the value of **Gross Retention**:

- For this risk only, the **Limit** for a quota share treaty is adjusted to the value of the new gross retention.
- For each surplus treaty for this risk only, the **Attachment point** and **Limit** scale proportionally based on their start and stop lines by using the following formulas:

Attachment = Start Line \* Gross Retention  
Limit = Stop Line \* Gross Retention
- As always, the amount of reinsurance provided is:

Amount of reinsurance = Limit - Attachment
- The ceded amount and proportional share are recalculated for all agreements in the list.

#### To modify the gross retention

1. In the policy transaction you created in “Adding Reinsurance to a Policy” on page 616, select **Tools → Reinsurance**.

2. Under **Reinsurable Risks**, select **Location 1**.

On the **Per Risk** tab, the **Gross Retention** defaults to \$1,000,000. This value is the amount of risk retained by the carrier prior to any amount ceded to the first surplus treaty. The **Attachment Point** of the **Surplus Treaty #1** is \$1,000,000.

The following table shows the values for the agreements:

Agreement	Attachment	Limit	Share %	Max Ceding	Ceded Risk	Prop %	Lines	Start Line	Stop Line
Quota share treaty		1,000,000	25	\$250,000	\$250,000	22.3214			
Surplus Treaty #1	\$1,000,000	\$4,000,000		\$120,000	\$120,000	10.7143	3	1	4
Surplus Treaty #2	\$4,000,000	\$15,000,000				0	11	4	15
Net Excess of Loss Treaty	\$500,000	\$750,000	100	\$250,000	\$250,000				

3. Cut the value of **Gross Retention** to 50% by entering 500,000 and clicking in another field such as **PML Reason**.
- The limit for the quota share treaty is set to the value of the new gross retention, \$500,000.
  - PolicyCenter scales the attachments and limits for the two surplus treaties.
  - The amount of reinsurance provided scales proportionally.
  - These changes do not affect the net excess of loss treaty.

Agreement	Attachment	Limit	Share %	Max Ceding	Ceded Risk	Prop %	Lines	Start Line	Stop Line
Quota share treaty		500,000	25	\$125,000	\$125,000	11.1607			
Surplus Treaty #1	\$500,000	\$2,000,000		\$620,000	\$620,000	55.3571	3	1	4
Surplus Treaty #2	\$2,000,000	\$7,500,000				0	11	4	15
Net Excess of Loss Treaty	\$500,000	\$750,000	100	\$250,000					

## Adding or Linking to a Facultative Agreement

A facultative agreement is placed on an individual case basis on a policy. In a policy or policy transaction, you can create a new facultative agreement.

You can also link to an existing facultative agreement created by selecting **Actions → New Facultative Agreement** in the **Reinsurance** tab. Or, the facultative agreement might have been created in an external reinsurance system. The facultative agreement attaches to one risk only. You cannot share the same facultative agreement across multiple risks. For example, you cannot attach the same facultative agreement to two different locations within the same policy.

### To add a new facultative agreement

- Navigate to the **Reinsurance** screen in a policy or quoted policy transaction.
- Under **Applicable Reinsurance**, select **Add Fac** → **Create New** and select one of the facultative agreement types. PolicyCenter displays the **Facultative** screen.
- Enter the details of the agreement.
- On the **Agreement Participants** tab, add at least one participant. A participant is a reinsurer participating in this agreement.
- Click **OK** to create the agreement.

**To link to an existing facultative agreement**

1. Navigate to the **Reinsurance** screen in a policy or quoted policy transaction.
2. Under **Applicable Reinsurance**, select **Add Fac** → **Link Existing**.  
PolicyCenter displays the **Add Agreements** screen.
3. Enter search criteria such as **Type** or **Status**, and click **Search**.
4. Select one or more facultative agreements in the **Search Results** and click **Add**.

**To remove a facultative agreement**

Add a check in the column on the far right of facultative agreement, and click **Remove Fac**.

**See also**

- “Facultative Agreements” on page 596

## Editing Ceding Parameters

The reinsurance plugin chooses reinsurance programs for each risk. The **Reinsurance** screen in a policy file displays the agreements in the programs. On this screen, you can modify which treaties cover the selected risk.

In the policy file, the **Per Risk** tab of the **Reinsurance** screen lists the reinsurance agreements in the selected program. In the column on the far right, each treaty has an **Edit** button.

**To edit ceding parameters**

1. Click the **Edit** button.  
PolicyCenter displays the **Edit Ceding Parameters** screen.
2. In the **Inclusion** drop-down list, select one of the following choices:
  - **Included** – Default. Include a risk in coverage for a treaty. This value does not appear in the **Inclusion** column.
  - **Excluded** – Exclude a risk from coverage by a treaty. For example, this risk does not fit the treaty for a reason that the system cannot determine automatically.
  - **Special Acceptance** – Make a special inclusion that cedes risk to a treaty even though it does not fit the normal criteria for inclusion under the treaty. Select this value if you have the reinsurer’s agreement to include this risk.

Facultative agreements are added or removed on a per risk basis as described in “Adding or Linking to a Facultative Agreement” on page 620. Therefore, facultative agreements do not have the **Edit** button.

## Reinsurance Management Screens

This topic describes PolicyCenter screens related to Reinsurance Management.

- “Treaty or Facultative Agreement Screen” on page 621
- “Reinsurance Program Screen” on page 628
- “Search Agreements Screen” on page 629
- “Search Programs Screen” on page 630
- “Reinsurance Screen in the Policy File” on page 630

## Treaty or Facultative Agreement Screen

Use the **Treaty** screen for entering information about a treaty. Use the **Facultative Agreement** screen for entering or viewing information about a facultative agreement.

## Fields

The following table provides descriptions of the fields on the Treaty screen.

Field	Description
Agreement Number	Data type: String  An identifier for the agreement, usually assigned by the reinsurer. In the default configuration, the number is entered by hand. You can customize PolicyCenter to generate the agreement number automatically. You can also customize PolicyCenter to use the agreement number when communicating with other systems about this agreement.
Name	Data type: String  Text description of the agreement for the user interface.
Type	Data type: typelist  The type of agreement. You can only select the type when creating a new treaty.  For treaties, values are: <ul style="list-style-type: none"><li>• Per Even Treaty</li><li>• Annual Aggregate Treaty</li><li>• Quota Share Treaty</li><li>• Surplus Treaty</li><li>• Excess of Loss Treaty</li><li>• Net Excess of Loss Treaty</li></ul> For facultative agreements, values are: <ul style="list-style-type: none"><li>• Proportional Facultative Agreement</li><li>• Excess of Loss Facultative Agreement</li><li>• Net Excess of Loss Facultative Agreement</li></ul>
Effective Date	Data type: Datetime  The date that coverage begins under the agreement.
Expiration Date	Data type: Datetime  The date that coverage ends under the agreement.  For validation, the <b>Expiration Date</b> must be greater than the <b>Effective date</b> . Validation does not use the time portion of this field because the time that reinsurance starts on a particular day can differ by jurisdiction or country. If this agreement is part of a program, then the time of reinsurance is consistent for all agreements in the program.
Status	Data type: Typelist  The status of the agreement. Values are: <ul style="list-style-type: none"><li>• Draft – Not yet finalized or ready for use.</li><li>• Active – Finalized and usable.</li><li>• Inactive – Either never made active or removed after making active, but not longer in active use. This status does not indicate that the agreement is past its expiration date.</li></ul>
Currency	In a multicurrency system, you can search by <b>Currency</b> . This drop-down list displays the currencies configured in the base application. For more information, see “Multicurrency Policies” on page 517.
Coverage	
Ceded Share (%)	Data type: XX.XXXX%  The percentage ceded to the reinsurer. This field applies to the following types of agreements: <ul style="list-style-type: none"><li>• Quota share treaties – Applies to the ceded share within the layer. No default value.</li><li>• Facultative proportional – Applies to the ceded share for the agreement (total risk minus any risk covered by an excess of loss agreement).</li><li>• Non-proportional agreements – Applies to the ceded share within the layer. The ceded share extends from the attachment point to the coverage limit. Default value is 100%.</li></ul> <b>Note:</b> This field does not apply to surplus treaties because they do not have a fixed value for ceded share.

Field	Description
Coverage Limit	Defines the upper bound on coverage.  <b>Note:</b> Does not apply to proportional facultative agreements.
Coverage Limit Indexed?	Data type: Boolean  Whether losses are subject to adjustment for inflation before being compared to the attachment point and limit of the agreement.  <b>Note:</b> Applies to non-proportional agreements.
Amount of RI	Data type: Money  Defines the maximum amount of reinsurance that can be recovered from the agreement.  PolicyCenter calculates the value of this field from the attachment point and coverage limit by using a formula: <ul style="list-style-type: none"><li>• For most types, Amount = (Coverage Limit - Attachment Point) * Ceded Share %</li><li>• For quota share, Amount = Coverage Limit * Ceded Share %</li></ul> <b>Note:</b> Does not apply to proportional facultative agreements. For proportional facultative agreements, the amount of reinsurance is equal to the Amount of Risk Ceded.
Amount of Risk Ceded	Data type: Money  Amount of risk ceded.  <b>Note:</b> Applies to proportional facultative agreements.
Attachment Point	Data type: Money $\geq 0$  The lower limit for the start of coverage. For a Surplus Treaty, this is the lower limit assuming the maximum retention.  <b>Note:</b> Does not apply to quota share or proportional facultative treaties.
Attachment Point Indexed?	Data type: Boolean  Default: false  Whether to adjust the loss costs by an inflation index before determining whether loss costs exceed the attachment point.  <b>Note:</b> Applies to non-proportional agreements.
Max Retention	Data type: Money $\geq 0$  Default: Attachment Point  The maximum underlying gross retention that the carrier can hold. This retention serves as the basis for the amount of reinsurance coverage provided.  For validation, the maximum retention must be less than or equal to the attachment point.  <b>Note:</b> Applies to surplus treaties.
Lines	Number of lines of coverage. PolicyCenter calculates this value by using the following formula:  $\text{(Coverage Limit} - \text{Attachment Point}) / \text{Max Retention}$  The number of lines is used to calculate the values of Attachment and Limit on the Per Risk tab in the policy file. For an example, see "Modifying the Gross Retention" on page 619.  <b>Note:</b> Applies to surplus treaties.
Start Line	PolicyCenter calculates this value by using the following formula:  $\text{Attachment Point} / \text{Max Retention}$  <b>Note:</b> Applies to surplus treaties.
Stop Line	PolicyCenter calculates this value by using the following formula:  $\text{Coverage Limit} / \text{Max Retention}$  <b>Note:</b> Applies to surplus treaties.

Field	Description
<b>Premium and Commissions</b>	
Min Deposit Premium	Data type: Money
	Indicates the minimum premium due for the contract regardless of any per risk calculation of ceded premiums. PolicyCenter does not use this field in the default configuration.
	<b>Note:</b> Applies to treaties.
Deposit Payment Schedule	Data type: Typelist
	The payment schedule for the deposit premium. Values are:
	<ul style="list-style-type: none"> <li>• Fully in advance</li> <li>• Quarterly in advance</li> </ul>
	PolicyCenter does not use this field in the default configuration.
	<b>Note:</b> Applies to treaties.
Payable on Written	Data type: Typelist
	When the ceded premiums and commission are payable. Values are:
	<ul style="list-style-type: none"> <li>• As Written – The ceded premiums and commissions are reported and considered payable as the written premium is recognized.</li> <li>• As Earned – The ceded premiums and commissions are considered payable as the underlying premium is earned.</li> </ul>
	PolicyCenter does not use this field in the default configuration.
Commission (%)	Data type: XX.XXXX%
	The commission that the carrier earns from the reinsurers for sending them reinsurance business.
	The commission is a percentage of the ceded premium. The commission for a participant to the agreement is calculated by using this commission percentage and the participant's commission rate.
	Enter 0 if there is no commission. For example, many non-proportional agreements do not pay a commission.
Set Differential Commission Rates	Click this button to:
	<ul style="list-style-type: none"> <li>• Add a Commission % column for each participant in the <b>Agreement Participants</b> tab.</li> <li>• Remove the <b>Commission (%)</b> field.</li> </ul>
	For additional information, see "Shared Reinsurance Agreements" on page 610.
Broker	Data type: Contact
	The broker, if any, who helped arrange this agreement.
Ceded Premium (flat amount)	Data type: Money
	For facultative agreements, the flat amount that is paid to the reinsurer rather than ceding a proportional percentage.
	<b>Note:</b> Applies to non-proportional facultative agreements. Applies to proportional facultative agreements if the value is greater than 0.
Set Differential Flat Premium	Click this button to:
	<ul style="list-style-type: none"> <li>• Add a Flat Premium column for each participant in the <b>Agreement Participants</b> tab.</li> <li>• Remove the <b>Ceded Premium (flat amount)</b> field.</li> </ul>
	For additional information, see "Shared Reinsurance Agreements" on page 610.
	<b>Note:</b> Applies to facultative agreements.
Ceding Rate (%)	Data type: XX.XXXX%
	A percentage per dollar of the underlying net premium that is ceded to the reinsurer.
	For validation, the ceding rate must be greater than or equal to 0.
	<b>Note:</b> Applies to non-proportional treaties.

Field	Description
Set Differential Ceding Rates	<p>Click this button to:</p> <ul style="list-style-type: none"> <li>Add a <b>Ceding Rate (%)</b> column for each participant in the <b>Agreement Participants</b> tab.</li> <li>Remove the <b>Ceded Rate (%)</b> field.</li> </ul> <p>For additional information, see “Shared Reinsurance Agreements” on page 610.</p> <p><b>Note:</b> Applies to non-proportional treaties.</p>
Markup (%)	<p>Data type: XX.XXXX%</p> <p>Additional amount, expressed as a percentage of ceded premium, to subtract from direct premiums when determining the net premium for ceding to other agreements.</p> <p><b>Note:</b> Applies to excess of loss facultative and net excess of loss facultative agreements.</p>
Total Cost	<p>Data type: Money</p> <p>The total cost of placing the reinsurance in terms of how much premium is subtracted from total premium to understand what premium applies to the remaining risk. Proportional agreements share the remaining risk.</p> <p>The value of this field is calculated as:</p> $(1 + \text{Markup}) * \text{Ceded Premium}$ <p><b>Note:</b> Applies to excess of loss facultative and net excess of loss facultative agreements.</p>
<b>Other Terms</b>	
Count Toward Total Limit	<p>Data type: Boolean</p> <p>Default: true</p> <p>If false, none of the total insured value on a risk is formally ceded to this treaty when determining how much risk has been ceded versus retained. Set to false for a treaty that provides coverage in the case of replacement cost error. For example, the actual loss could exceed the notional total insured value. Therefore the treaty has no effect on how much of the known total insured value has been ceded or retained.</p> <p>This limit is not counted when determining the maximum amount of insurance that any one risk can have.</p> <p><b>Note:</b> Applies to all per risk agreements. In the default configuration, this field does not apply to annual aggregate or per event treaties.</p>
Notification Threshold	<p>Data type: Money <math>\geq 0</math></p> <p>Default: 0</p> <p>Notify the reinsurer if an individual large loss exceeds this threshold.</p> <p><b>Note:</b> Applies to per risk agreements.</p>
Comments	<p>Data type: Text</p> <p>A text field for the carrier to add notes such additional terms that the system does not use but which are important to document to fully describe the agreement.</p>
Gross Net Premium Basis	<p>Data type: Typelist</p> <p>When calculating how much premium to cede to a treaty, the <i>gross net premium</i> is the premium you are starting with. For example, proportional treaties get a share of the premium <i>net of all excess of loss treaties</i>. The net premium coming in is the <i>gross net premium</i> and premium coming out (after ceding to proportional treaties) is the <i>net net premium</i>. The value of this field is used to determine what premium, if any, to net out prior to calculating premium for this treaty.</p> <p>Determine the basis of the gross net premium by using one of the following methods:</p> <ul style="list-style-type: none"> <li><b>Gross Premium</b> – Prior to netting out any other ceded premiums (including proportional cedings)</li> <li><b>Net of proportional</b> – Net premiums after ceding to proportional agreements but prior to any other non-proportional cedings</li> <li><b>Net of per risk</b> – Premiums after all cedings to per risk agreements but prior to any cedings to aggregate agreements</li> <li><b>Net of all per event</b> – Premiums after all cedings to per event agreements</li> <li><b>Net of all prior</b> – Premiums after all cedings</li> </ul> <p><b>Note:</b> Applies to all non-proportional agreements.</p>

Field	Description
Calculate Ceded Premium	<p>Data type: Boolean</p> <p>Premiums are ceded based on either written value or earned value for the reinsured period. This field is intended to control whether PolicyCenter calculates ceded premiums on a per policy basis. It is common to calculate ceded premiums for non-proportional agreements on a whole of class basis rather than on a per policy basis. For example, a carrier often calculates ceded premiums quarterly based on all earned premium for a whole set of policies. If ceded premiums are calculated on a whole of class basis, then there is no reason to calculate those ceded premiums in PolicyCenter. In this case, you can set this value to No.</p> <p>In this version of PolicyCenter, ceded premiums are not calculated even if you set this value to Yes. However, you can configure PolicyCenter to calculate these premiums.</p> <p><b>Note:</b> Applies to non-proportional agreements.</p>
Loss Attachment Basis	<p>Determines whether reinsurance coverage is based on the date of loss or the policy period effective date.</p> <p>Values are:</p> <ul style="list-style-type: none"> <li>• <b>Loss Date Attachment (Earned Premium)</b> – Ceded premiums are paid based on earned premiums that fall within the treaty period.</li> <li>• <b>Loss Date Attachment (Written Premium)</b> – Ceded premiums are paid based on written premiums that fall within the treaty period.</li> <li>• <b>Policy Attachment</b> – This value is available for excess of loss and net excess of loss treaties.</li> </ul> <p><b>Note:</b> Applies to non-proportional agreements.</p>

## Agreement Participants Tab

On the **Agreement Participants** tab, add all reinsurers participating in this agreement. Among all participants, the total risk share must equal 100%.

The following table provides descriptions of the fields on the **Agreement Participants** screen.

Field	Description
Participant	<p>Data type: Contact</p> <p>The name of the reinsurer. This field has a link to the contact information for the reinsurer.</p>
Risk Share %	<p>Data type: XXX.XXXX%</p> <p>Enter the participant's share of any losses to the agreement. When you add a new row, this value defaults to the remaining amount to get to 100%.</p>
Ref #	<p>Data type: String</p> <p>Enter the agreement identifier provided by the reinsurer. This field is similar to a carrier's policy number.</p>
Ceding Rate (%)	<p>Data type: XX.XXXX%</p> <p>Appears when you click the <b>Set Differential Ceding Rates</b> button.</p> <p>If multiple participants share a non-proportional treaty, then they each negotiate a rate for their participation.</p> <p><b>Note:</b> Applies to non-proportional treaties.</p>
Premium Share %	<p>Data type: XX.XXXX%</p> <p>Appears when you click the <b>Set Differential Ceding Rates</b> button.</p> <p>PolicyCenter calculates each participant's share of the overall ceded premium, so that ceded premiums calculated for the treaty as a whole can be divided among the participants. The formula for premium share is:</p> $\text{Participant's Ceding Rate} / \text{Overall Ceding Rate}$ <p><b>Note:</b> Applies to non-proportional treaties.</p>

Field	Description
Commission %	<p>Data type: XX.XXXX%</p> <p>Appears when you click the <b>Set Differential Commission Rates</b> button.</p> <p>The commission that will be paid to each participant. Defined as a percentage of their share of ceded premiums.</p>
% of Total Commissions	<p>Data type: XX.XXXX%</p> <p>Appears when you click the <b>Select Differential Commission Rates</b> button.</p> <p>How the participants share the commission.</p> <p>PolicyCenter calculates this field by using this formula:</p> $\text{(Risk Share * Commission Rate) / Agreement Overall Commission Rate}$ <p>The formula for the Agreement Overall Commission Rate is:</p> $\text{sum of (Risk Share * Commission Rate) for each participant}$
Flat Premium	<p>Data type: Money</p> <p>Appears when you click the <b>Set Differential Flat Premium</b> button.</p> <p>Enter the premium amount that paid to each participant.</p> <p><b>Note:</b> Applies to facultative agreements.</p>

### Commissions

Commission rates can be uniform across all participants or can be set individually. You can toggle these two modes by clicking the **Set Differential Commission Rates** button. If you choose to set the rates individually, PolicyCenter automatically calculates the commission rate of the agreement. If you choose to set differential commission rates, you enter the commission rate for each participant in the **Commission %** column on the **Participants** tab.

The overall commission is based on each participants share percentage and commission rate.

### Applies To Tab for Agreements

You can add and remove reinsurance coverage groups on the **Applies To** tab. The **Remove** button appears if you select one or more reinsurance coverage groups in the agreement.

The default configuration contains the following reinsurance coverage groups:

- Auto Liability
- Auto PD
- Liability
- Property
- Workers Comp

**Note:** Reinsurance coverage groups apply to treaties.

### See also

- “Reinsurance Coverage Groups for Treaties” on page 595

## Reinsurance Program Screen

The following table provides descriptions of the fields that appear on the Reinsurance Program screen.

Field	Description
Name	Data type: String Text description of the program for the user interface.
Effective Date	Data type: Datetime The date that coverage begins under the program. The effective dates for all included agreements must exactly match the effective range of the program.
Expiration Date	Data type: Datetime The date that coverage ends under the program. For validation, the <b>Expiration Date</b> must be greater than the <b>Effective date</b> .
Status	Data type: Typelist The status of the program. Values are: <ul style="list-style-type: none"><li>• Draft – Not yet finalized or ready for use.</li><li>• Active – Finalized and usable.</li><li>• Inactive – Either never made active or removed after making active, but not longer in active use. This status does not indicate that the program is past its expiration date.</li></ul>
Currency	In a multicurrency system, the currency of the program. For more information, see “Multicurrency Policies” on page 517.
Target Max Retention	Data type: Money Enter the target retention value for all risks covered under this program. The target retention value is the maximum risk amount you wish to retain. PolicyCenter creates an underwriting issue if reinsurance coverage is insufficient for a given risk. The coverage is insufficient because the risk retained is greater than the target maximum retention. For more information about the underwriting issue, see “Reinsurance Net Retention Rule” on page 532 in the <i>Configuration Guide</i> .
Single Risk Maximum	Data type: Money The maximum risk that the program covers. Increasing the <b>Single Risk Maximum</b> may require a corresponding increase in the <b>Target Max Retention</b> .
Test Retention	When you click the <b>Test Retention</b> button, PolicyCenter calculates the <b>Implied Retention</b> value for the program. The system calculates how risk is ceded and how is retained for a notional risk with Total Insured Value = Single Risk Maximum. Click this button to verify that the program actually results in the net retention you expect for a risk whose TIV = Single Risk Maximum.
Implied Retention	Data type: Money PolicyCenter calculates this value when you click the <b>Test Retention</b> button. The value is the retention for all per risk agreements in the program.

## Treaties Tab for Programs

The **Treaties** tab displays treaties that the program contains. The **Treaties** tab has two panels.

The **Per Risk** pane displays the following types of treaties:

- Quota Share Treaty
- Surplus Treaty
- Excess of Loss Treaty
- Net Excess of Loss Treaty

The Per Risk pane displays proportional agreements, ordered by their attachment points in increasing order. Quota share treaties always appear first because they have an attachment point of 0. Surplus treaties appear next in increasing attachment point order. After proportional agreements, the Per Risk pane displays non-proportional agreements. The pane displays excess of loss treaties followed by net excess of loss treaties. Both types of treaties are appear in increasing attachment point order.

The Aggregate pane displays the following types of treaties:

- Per-Event Treaty
- Annual Aggregate Treaty

The Aggregate pane displays all aggregate agreements, with per-event treaties before annual aggregate treaties. Within each category, treaties are in increasing attachment point order.

The search only returns treaties with the same currency as the program.

## Applies To Tab for Programs

The Applies To tab displays Included Coverage Groups for the current program.

To add a reinsurance coverage group, click Add and select a reinsurance coverage group from the drop-down menu. Select the check box next to one or more reinsurance coverage groups and click Remove to remove reinsurance coverage groups.

## Search Agreements Screen

The following table provides descriptions of the fields on the Search Agreements screen.

Field	Description
Agreement Number	The agreement number must match exactly.
Name	Enter one or more characters at the beginning of the name.
Effective Date	Values are: <ul style="list-style-type: none"><li>• &lt;none selected&gt;</li><li>• Coming Year – Find agreements with effective date starting at the current date through one year from this date.</li><li>• Last Year – Find agreements in programs that are in effect on the current date and back one year from this date.</li><li>• Custom Dates</li></ul> If you select Coming Year or Last Year, the From field is set to the current date. The To field is set to one day before one year forwards or backward from the current date. If you select Custom Dates, you can set the From and To fields.
From	PolicyCenter fills in the effective date of the agreement.
To	PolicyCenter fills in the expiration date of the agreement.
Type	Select the type of agreement from the drop-down list.
Arrangement	Values are: <ul style="list-style-type: none"><li>• &lt;none selected&gt;</li><li>• Treaty</li><li>• Facultative</li></ul>
Coverage Group	Select a reinsurance coverage group from the drop-down list.
Status	Select a status from the drop-down list. Values are: <ul style="list-style-type: none"><li>• &lt;none selected&gt;</li><li>• Draft</li><li>• Active</li></ul>
Currency	In a multicurrency system, specify the currency of the agreement. For more information, see “Multicurrency Policies” on page 517.

## Search Programs Screen

The following table provides descriptions of the fields on the Search Programs screen.

Field	Description
Name	Enter one or more characters at the beginning of the name.
Coverage Group	Select a reinsurance coverage group from the drop-down list.
Status	Select a status from the drop-down list. Values are: <ul style="list-style-type: none"> <li>• &lt;none selected&gt;</li> <li>• Draft</li> <li>• Active</li> </ul>
Currency	In a multicurrency system, specify the currency of the program.
Effective Date	Values are: <ul style="list-style-type: none"> <li>• &lt;none selected&gt;</li> <li>• Coming Year</li> <li>• Last Year</li> <li>• Custom Dates</li> </ul> <p>If you select Coming Year or Last Year, the From field is set to the current date. The To field is set to one day before one year from the current date.</p> <p>If you select Custom Dates, you can set the From and To fields.</p>
From	PolicyCenter fills in the effective date of the agreement.
To	PolicyCenter fills in the expiration date of the agreement.

## Reinsurance Screen in the Policy File

In the policy file, the Tools → Reinsurance screen displays reinsurable risks on the policy. The following table describes the fields on this screen.

Field	Description
Ceded Premium Recalc Reason	This field appears if you edit the reinsurance for a policy directly in the policy file rather than as part of processing a policy transaction.  Click Edit to display this field. Enter a reason for recalculating the ceded premium. If this value results in a change to ceded premium, this reason appears on the View Ceded Premiums screen for all views except All Transactions. PolicyCenter displays the reason after running the Premium Ceding batch process.
<b>Reinsurable Risks</b>	
View As Of	This drop-down list shows the ranges of dates for programs that apply during the policy period. For loss attachment agreements, PolicyCenter selects the agreement that applies at the loss date rather than the policy period start date.
Risk	Risks are typically defined at the level of a single location (for owned property) or at the level of the whole policy (for liability). There is one risk for each combination of level and reinsurance coverage group.  For example, you might have two property risks on a location if there is one risk for normal property coverages and a separate risk for terrorism coverage. Terrorism coverage is often reinsured in a different way.
Coverage Group	The reinsurance coverage group of the risk.
TIV/Sum Insured	The total insured value, sometimes referred to as <i>sum insured</i> . The maximum amount for any single risk that could be paid for all coverages that are part of the risk.
Probable Max Loss %	Enter the probable maximum loss as a percentage of total insured value.

Field	Description
PML Amount	<p>The probable maximum loss. Usually, PML equals TIV, but may be less if <b>Probable Max Loss %</b> is less than 100.</p> <p>You can use the <b>PML Amount</b> to adjust the amount of risk used for allocating reinsurance. Specify the <b>PML Amount</b> if it is extremely unlikely that there will be a loss as large as the TIV.</p> <p>For example, a policy has multiple buildings at the same location. However, large distances separate the buildings. Therefore, it is very unlikely that the same fire will destroy all buildings.</p>
<b>Applicable Reinsurance</b>	
Per Risk tab	
Total Risk	The total risk. This risk is the total insured value or PML, if adjusted by <b>Probable Max Loss %</b> .
Covered by XOL	The amount of risk covered by all excess of loss agreements.
Shared Among Prop	The amount of risk shared among proportional agreements. This value is <b>Total Risk</b> minus <b>Covered by XOL</b> .
Gross Retention	<p>The amount of risk retained by the carrier prior to any amount ceded to the first surplus agreement. If there is a quota share treaty attached to this risk, this value defaults to the <b>Coverage Limit</b> of the quota share treaty. If this risk does not have a quota share treaty, then the value defaults to the <b>Max Retention</b> of the first surplus treaty. Specify the <b>Max Retention</b> on the <b>Reinsurance → Treaty</b> screen.</p> <p>You can modify the gross retention. However, the value must be less than or equal to the <b>Max Retention</b> of the first surplus treaty. For an example, see “Modifying the Gross Retention” on page 619.</p>
Retained Prop Share (%)	The percentage of risk retained by the carrier of the total shared among proportional agreements.
Prop Retention	The amount of risk retained by the carrier. This amount is the <b>Retained Prop Share (%)</b> expressed as an amount.
Covered By NXOL	The amount of risk covered by all net excess of loss agreements.
Net Retention	The net amount of risk retained by the carrier after ceding risk to all per risk reinsurance agreements.
Target Max Retention	<p>The target maximum retention. This retention is the value of <b>Target Max Retention</b> on the <b>Reinsurance Program</b> screen.</p> <p>If the carrier's net retention on the current risk exceeds this value, PolicyCenter creates an underwriting issue. An underwriter can approve an exception, put in place a facultative agreement, or decline the policy.</p> <p>For more information about the underwriting issue, see “Reinsurance Net Retention Rule” on page 532 in the <i>Configuration Guide</i>.</p>
Fac RI Needed	Facultative reinsurance needed. If the <b>Target Max Retention</b> is exceeded, this is the overage. The overage is the amount of risk that facultative reinsurance could cover to bring the actual retention down to the <b>Target Max Retention</b> .
<b>Per Risk and Aggregate fields</b>	
Agreement Number	The agreement number.
Name	The agreement name.
Type	The type of agreement.
Attachment	The attachment point.
Limit	The coverage limit.
Share %	The ceded share percentage. This percentage is the <b>Ceded Share (%)</b> on the <b>Treaty or Facultative Agreement</b> screen.
Max Ceding	Per risk only. The maximum amount of risk that can be ceded.

Field	Description
Ceded Risk	<p>Per risk only. The amount of risk that was ceded. In the default configuration, Ceded Risk is always set to the value of Max Ceding.</p> <p>In some cases, the carrier agrees to cede less risk than the maximum amount based on the characteristics of an individual risk. In this case, ceded risk can be less than Max Ceding. For more information, see "Reinsurance Configuration Plugin" on page 467 in the <i>Integration Guide</i>.</p>
Prop %	<p>Per risk only. For each proportional treaty, its percentage of the Shared Among Prop. This value is calculated according to the following formula:</p> $\text{Ceded Risk} / \text{Shared Among Prop}$
Inclusion	<p>Whether or not to include a risk in coverage by a treaty. This field has the following values:</p> <p><b>Included</b> – Default. Include a risk in coverage for a treaty. This value does not appear in the Inclusion column.</p> <p><b>Excluded</b> – Exclude a risk from coverage by a treaty. For example, this risk does not fit the treaty for a reason that the system cannot determine automatically.</p> <p><b>Special Acceptance</b> – Make a special inclusion that cedes risk to a treaty even though it would not normally.</p>
Projected	This field displays the value Projected if PolicyCenter selected an agreement from a draft program or an active program in a prior year. For more information, see "How PolicyCenter Selects a Projected Program" on page 605.

---

part VIII

# PolicyCenter Administration



# Security: Roles, Permissions, and the Community Model

Security in PolicyCenter is managed through roles, permissions, and producer codes. This security makes the application flexible, robust, and keeps your information protected. The PolicyCenter default application contains a set of roles that perform the policy tasks in most organizations. To perform these tasks, a user must be assigned a role with the appropriate permissions. Typically, the Superuser role is granted all permissions and is responsible for granting permissions to other roles. Once the roles are configured, then each PolicyCenter user is assigned a specific role that relates to the tasks to be performed. Besides role-based security, PolicyCenter also provides data-based security which is managed through producer codes and organizations.

PolicyCenter also allows you to control access to documents and notes through access permissions.

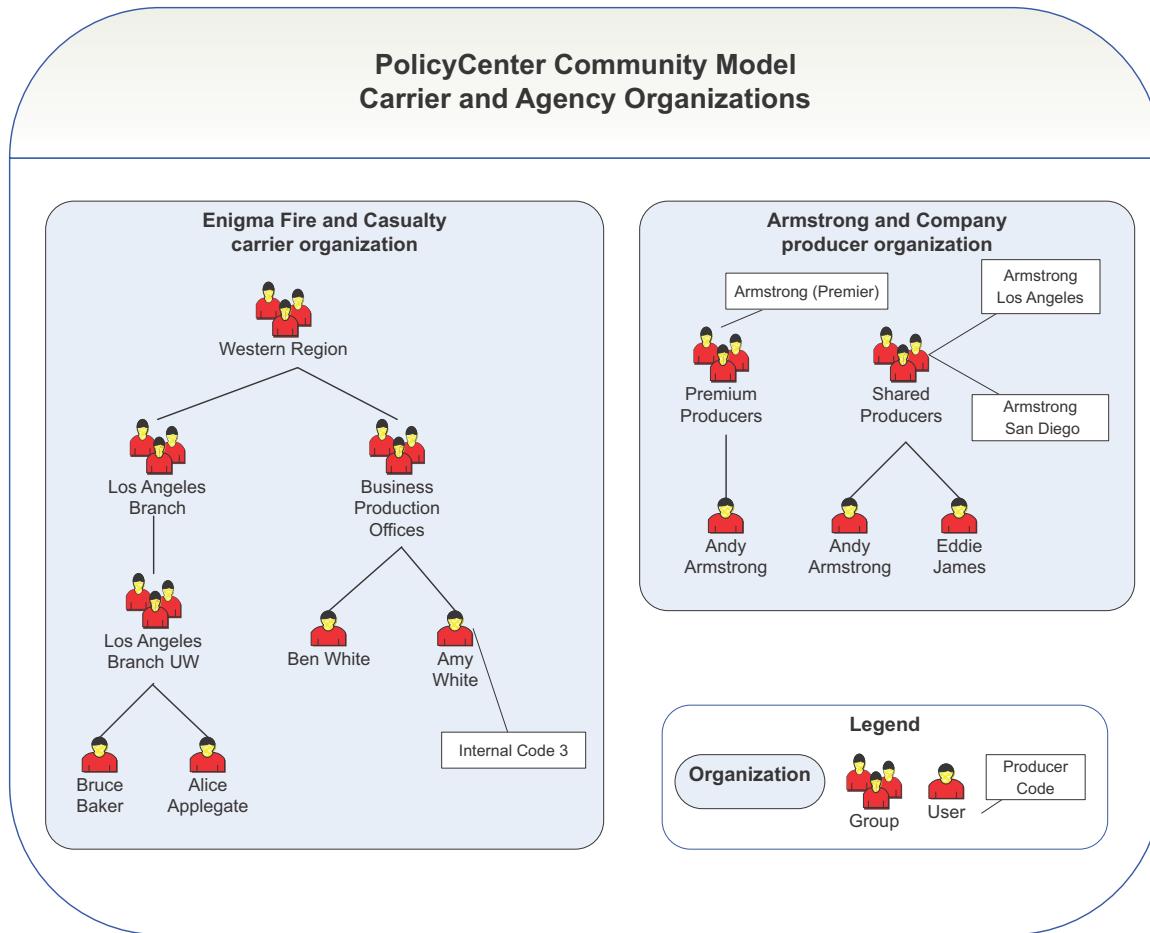
This topic includes:

- “Community Model Overview” on page 635
- “Security Overview” on page 637
- “Producer of Record and Producer of Service” on page 644
- “Managing the PolicyCenter Community” on page 645
- “Security Object Models” on page 646
- “Working with Security” on page 647
- “Security and Configuration Scenarios Related to Producer Codes” on page 649
- “Security Dictionary” on page 651
- “Access Control for Documents and Notes” on page 652

## Community Model Overview

PolicyCenter uses a *community model* to organize users into organizations. The organization of the carrier is the internal organization. There can be only one internal organization. There can be one or more external organizations, such as an agency. Users, groups, and producer codes all belong to a single organization. Each organization

has groups underneath it. Groups can also have subgroups. The following illustration shows the group structure in the Enigma Fire and Casualty carrier organization and the Armstrong and Company agency organization.



There are two types of users:

- **Internal** – An internal user is an employee of the carrier. An internal user can potentially see groups and permissions of external users. In the illustration, the users in the Enigma Fire and Casualty carrier organization are internal users.
- **External** – An external user is a person outside the carrier organization who needs to access PolicyCenter data. External users are typically producers who work for an agency. External users must be associated with an organization. In the illustration, the users in the Armstrong and Company agency are external users. Armstrong producers are producers on Enigma Fire policies, but may also be producers on policies for other carriers. You can use external users and organizations to model either *captive agencies* that only sell business for Enigma, or *external agencies* that sell business to Enigma and other carriers.

Within an organization, users can be assigned to one or more groups. In the illustration above, the user, Andy Armstrong, belongs to the Premium Producers and the Shared Producers groups.

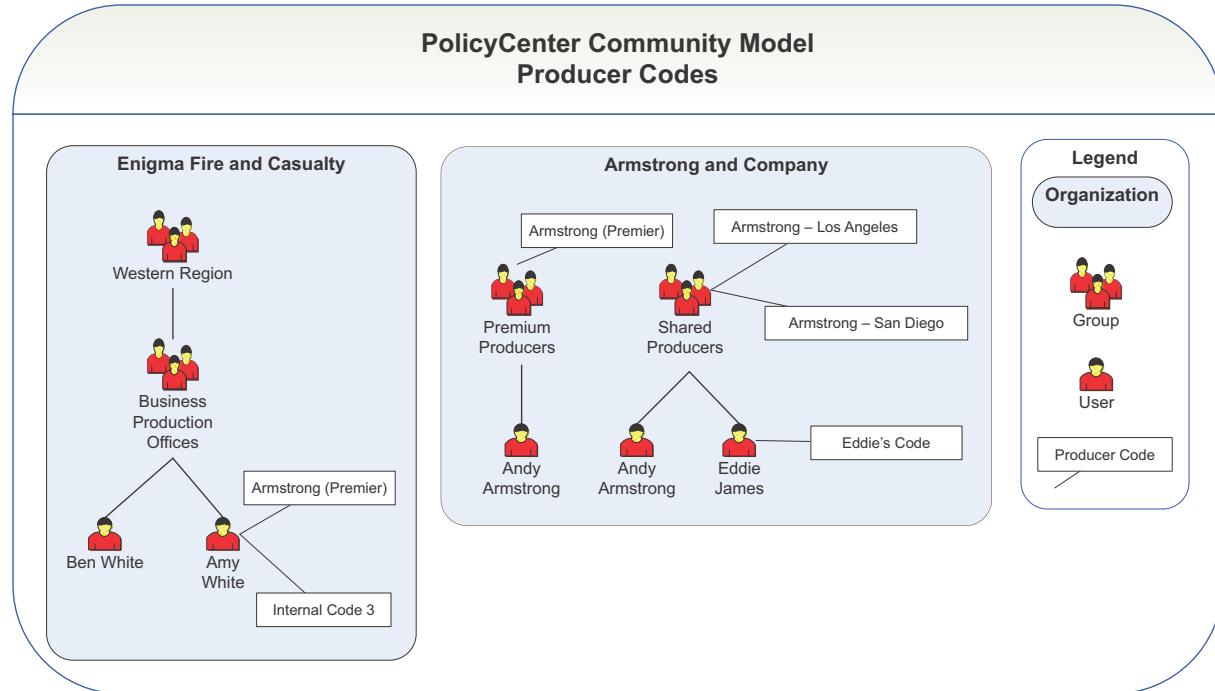
## Producers in the Community Model

A *producer code* is a unique identifier assigned by the carrier to accurately track the agent or agency responsible for a policy or account. The producer code also identifies the sales agent who sold the policy.

Producer codes can be assigned to groups within the organization or assigned directly to users. Consequently, users have a list of producer codes that apply to them, either explicitly, or through one of the user's groups.

External users may only have producer codes from their organization assigned to them. Internal users may have any producer codes assigned to them, from any organization. In the following illustration, the internal user, Amy White, who works at the carrier organization, has the Armstrong (Premier) producer code assigned to her.

In the following illustration, the user, Eddie, inherits producer codes from the Shared Producers group and has one producer code assigned directly to him. The user, Andy Armstrong, inherits producer codes from the Premium Producers and Shared Producers groups.



Carriers may think of producer codes as following their organization or group hierarchy, so assigning producer codes to groups can be a simple way to administer producer codes.

## Security Overview

PolicyCenter provides the following types of security:

- **Role-based Security** – Grants permission to perform actions such as create a submission or edit an account.
- **Data-based Security for Accounts and Policies** – Defines a user’s access to data.
- **Data-based Security for the Community Model** – Limits access to users, groups, organizations, and producer codes.
- **System and Application Permissions** – Fine tune user access.
- **Security Restrictions Using the Status Field**

### Role-based Security

*Role-based security* provides permission to perform an action based on a user’s role. Creating a submission and editing an account are examples of actions. Producers or auditors are examples of roles. Permissions and roles enforce role-based security.

### Permissions

A *permission* is a granular task or ability to see or do something within PolicyCenter, such as create submissions or edit accounts. Generally, permissions are grouped together depending on usage. For example an underwriter

would have the set of permissions that are necessary to perform underwriting work. This set of permissions define the user role of an underwriter. By grouping permissions into roles, a user's authority can be precisely defined by a few assigned roles, rather than by a much larger list of permissions. Some permissions govern access to entire sections of the application. For example, only users with the Superuser or Rule Admin roles are granted the ability to access internal server and debugging tools. Other permissions govern more granular actions, such as the ability to view, edit, create, or bind a submission.

## Roles

A *role* is a named collection of permissions and typically, maps to a job function or job title. For example, the *producer* role contains the set of permissions appropriate for someone who is a producer. For example, this role might have the *create submissions* or *edit accounts* permissions, but not the *create users* or possibly even the *issue submissions* permissions. Similarly, a *producer clerical* role might have only *create submissions* and not *edit accounts*. A user can have one or more roles, and must have at least one. The user is granted all of the permissions contained in any of the assigned roles. Roles provide the basic security that governs which actions the user can take within PolicyCenter.

## Data-based Security for Accounts and Policies

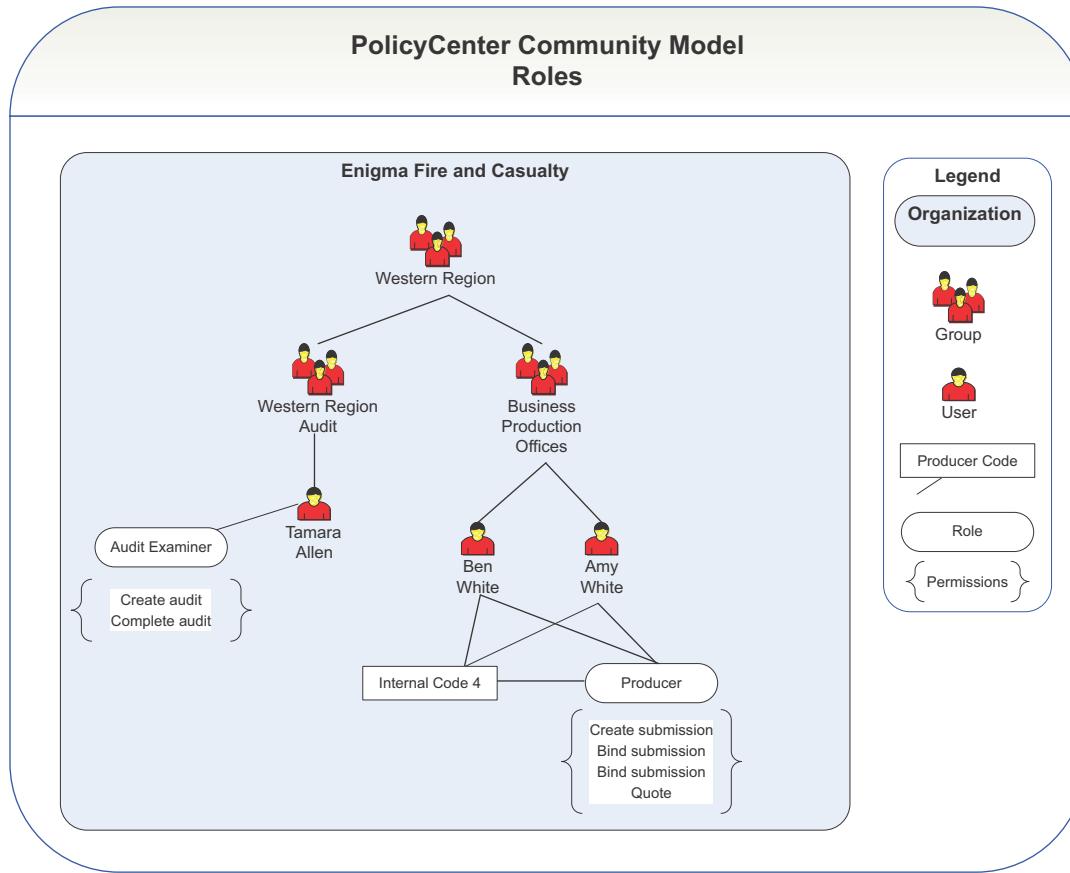
*Data-based security for accounts and policies* defines exactly which accounts and policies you have access to. PolicyCenter manages this security through producer codes and organizations. A carrier assigns producer codes to producers. Producers can be internal or external. You can think of producer codes as enforcing the agreement between the producer and the carrier that determines exactly what a producer can do on an account or policy. A carrier can attach permissions to producer codes, then attach producer codes to users and groups. This arrangement allows a carrier to:

- Define a user's access to accounts and policies.
- Define a user's actions on accounts and policies.
- Track the performance of a producer or other user.

## Roles are Groups of Permissions

*Roles* are groups of permissions. You can assign roles to users. You can also assign roles to producer codes. The roles assigned to a producer code customize what a user with that producer code can do. “Role-based Security” on page 637 describes how to assign roles to users.

The following illustration shows how producer codes and users are attached to roles. Each role has permissions associated with it.



For example, a PolicyCenter application might have roles for an audit examiner and a producer. The audit examiner has permissions to complete and create an audit. The producer has permissions to create submissions, quote, and bind policies.

## Roles Have Types

The role *type* specifies whether that role can be assigned to users, producer codes, or both. PolicyCenter has the following role types:

- **User Role** – Can be assigned only to users.
- **Producer Code Role** – Can be assigned only to producer codes.
- **User Producer Code Role** – Can be assigned to either users or producer codes.

**Note:** The User Producer Code Role is not functionally necessary, but allows you to have one role that you can assign to both users and producer codes. It would require two roles to assign the same permissions to both producer codes and users if this type did not exist.

## Producer Code Security and Policies

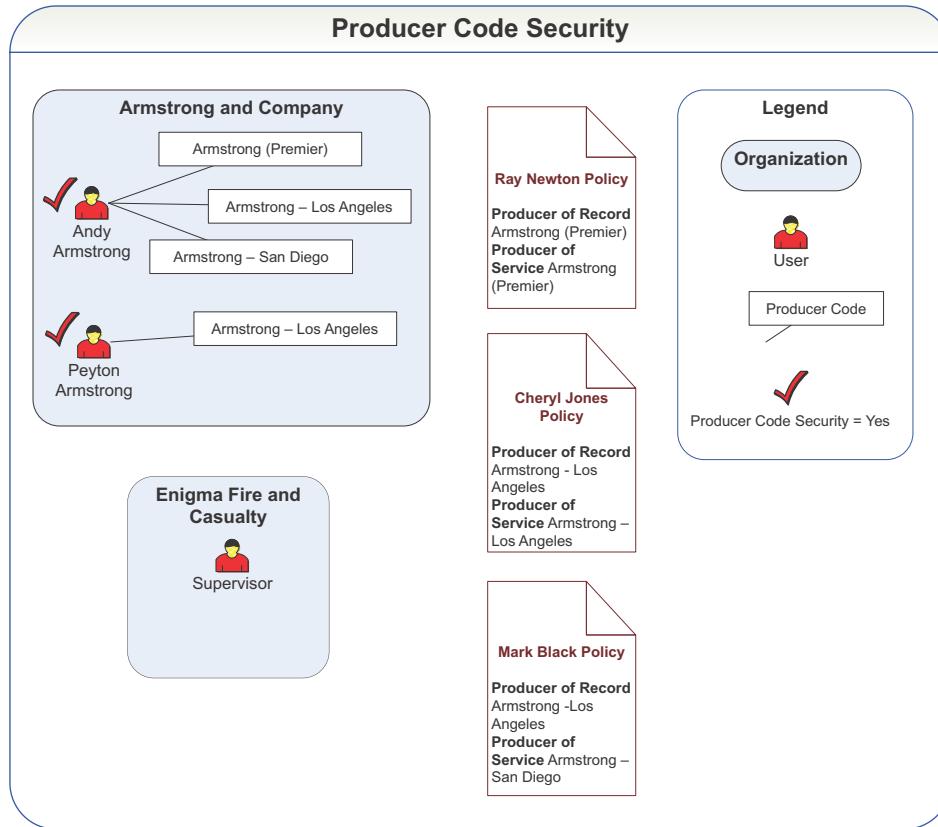
Producer code *roles* allow you to create groups of permissions and attach them to producer codes. You can then assign the producer codes to users, to policies, or accounts. Users who have producer code security can only use their role permissions against policies that have a producer code matching one of their own producer codes. Only the permissions assigned to the producer code that matches can be used against that policy, even if the user has a larger set of permissions available through user roles. A permission must exist on both the user role and the matching producer code role in order for it to be used with the policy.

You can turn producer code security on or off for any given user. This security is typically turned on for a producer. This security might not be turned on for underwriters or auditors, thus allowing them to work on any policy. For instructions on how to change this setting, see “Turning On Producer Code Security” on page 648..

In the following illustration, the producer of service is Armstrong (Premier) for the Ray Newton policy. Andy Armstrong has that producer code and has producer code security turned on. Therefore, he can use the permissions provided with the Armstrong (Premier) producer code on the Ray Newton policy.

Peyton Armstrong has producer code security turned on. He can use the permissions provided with the Armstrong - Los Angeles producer code to work on the Cheryl Jones policy. Peyton Armstrong cannot work on the Mark Black policy because the producer of service is Armstrong - San Diego.

The Supervisor has producer code security turned off. Therefore, she can work on any policy.



For more information about the producer of service and producer of record, see “Producer of Record and Producer of Service” on page 644.

## Standard Roles in PolicyCenter

With the appropriate permissions, you can access the list of roles by navigating to **Administration** → **Users & Security** → **Roles**. The following table lists the standard roles in PolicyCenter:

Role name	Role type	Description
Audit Examiner	User Role	Permissions for an audit examiner who processes audits received from the premium auditors. Typically, an audit examiner has visibility to underwriting information, but only write access to audit information.
Community Admin	User Role	Permissions for administration of the PolicyCenter community model (organizations, groups, users, and other administrative data).

Role name	Role type	Description
Integration Admin	User Role	Permissions for an integration administrator. This role typically performs duties such as SOAP administration.
Loss Control	User Role	Permissions for a loss control user. This user typically inspects businesses for underwriting purposes. In the default configuration, this role is a placeholder.
Premium Auditor	User Role	Permissions for a premium auditor who performs audits. A premium auditor is usually an internal employee who is not physically located at the carrier's office. Typically, a premium auditor has the same permissions as the Audit Examiner role, but cannot manually price the policy, bind, or complete the audit.
Processor	User Role	Permissions for a processor who typically has responsibility for policy management and production activities, but who is not part of the underwriting or audit organization. In the default configuration, this role is a placeholder.
Producer	User Producer Code Role	Permissions for a producer. This role is for an agent or broker. Has view access to underwriting information, but only write access to a limited number of jobs.
Producer Code - Basics	Producer Code Role	A subset of the overall permissions for a producer. Can be assigned to a producer code to limit the rights available when using that producer code.
Producer Code - Cancellations	Producer Code Role	A subset of the overall permissions for a producer with permissions for cancellation jobs. Can be assigned to a producer code to limit the rights available when using that producer code.
Producer Code - Policy Changes	Producer Code Role	A subset of the overall permissions for a producer with permissions for policy change jobs. Can be assigned to a producer code to limit the rights available when using that producer code.
Producer Code - Renewals	Producer Code Role	A subset of the overall permissions for a producer with permissions for renewal jobs. Can be assigned to a producer code to limit the rights available when using that producer code.
Producer Code - Submissions	Producer Code Role	A subset of the overall permissions for a producer with permissions for submission jobs. Can be assigned to a producer code to limit the rights available when using that producer code.
Producer Clerical	User Role	Base permissions for a clerical worker of an external producer organization. Has access to only limited policy information.
Reporting Admin	User Role	Permissions for a user administering reports. In the default configuration, this role is a placeholder.
Rule Admin	User Role	Permissions for a user administering system rules.
Superuser	User Role	Superuser with full permissions. Has every permission available in the system.
Underwriter	User Role	Permissions for an underwriter.
Underwriter Assistant	User Role	Permissions for an underwriter assistant. Typically, these permissions are more limited than for an underwriter.
Underwriting Supervisor	User Role	Base permissions for an internal carrier underwriting supervisor. Also includes additional management permissions.
User Admin	User Role	Permissions for user administrator, including granting or revoking of roles. Assign this role to external users responsible for administering the users within their organization.

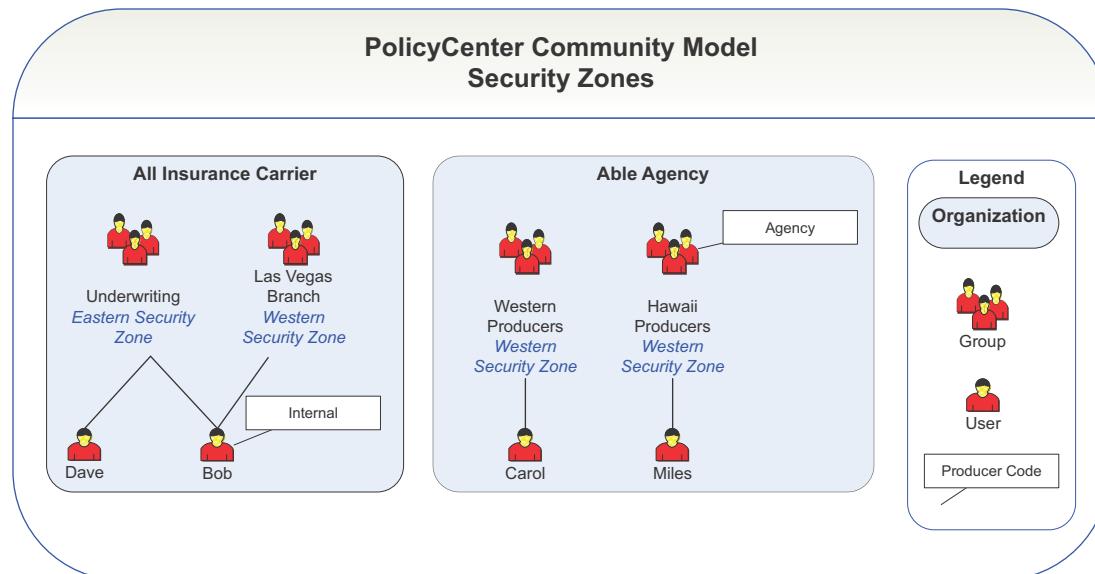
Administrators can assign roles to users. They can also create new roles and add or remove permissions from a role through the **Administration** tab. To learn more, see “Working with Security” on page 647.

## Account-level Security

Producer code security applies at the account, policy, and job levels. An account contains an array of producer codes which determines access to the account. By default, this array is comprised of all the producer codes of service from the bound policies on the account. However, you can modify this logic with Gosu in Studio.

## Data-based Security for the Community Model

PolicyCenter also has *data-based security for the community model* that limits the current user's access to users, groups, organizations, and producer codes. Most internal users have limits on their access to users, groups, organizations, and producer codes. These limits are defined by the *security zones* associated with the user. The security zone is a field defined for each group and organization. The security zones for a user are the security zones for all the groups that the user is assigned to. The security zone for a producer code is the security zone for the organization that the producer code belongs to.



For example, user Bob belongs to the All Insurance Carrier organization. Internal user Bob belongs to two groups that are in the Eastern and Western security zone, respectively. The Able Agency organization is in the Western security zone. The users in this agency organization are external users. Because Miles is an external user, he can access the users and producer codes in the Able Agency organization. Miles cannot access the Las Vegas Branch, even though it is in the Western Security zone. Because Bob and Able Agency are in the Western security zone, Bob can find the Able Agency organization, the user Carol, and the Agency producer code in a search. On the other hand, because Dave is in the Eastern security zone, he cannot access the Able Agency organization, its producer code, or users.

These restrictions by security zone apply when an internal user is searching for either internal or external groups, users, producer codes, or organizations.

An internal user who has been granted the **View All Users** permission can access any user, group, producer code, or organization, regardless of security zone.

Access for external users is further restricted. An external user can only see their own organization and the users, groups and producer codes that belong to their own organization. The security zone field for a group or organization is also read-only for external users. Consequently, a delegated administrator for an external organization is not able to view or modify anything related to other organizations. The delegated administrator is also unable to change the visibility of their own organization.

In the illustration above, the users in the Able Agency are external users. They cannot see into the All Insurance Carrier organization.

The access restrictions described above apply to searches and the ability to view the details of a user, group, organization, or producer code. However, just because a user can find and view a producer code does not mean that they can use the producer code for action. That is, the user may not be able to select the producer code when creating a new account. Assume that producer code security is turned on for a user. The user can only select producer codes for action if the producer code is associated with the user or with one of the groups the user belongs to. If producer code security is turned off for a user, that user can select any producer code that they can find through search.

## System and Application Permissions

In PolicyCenter, you can further control what a user can or cannot see or do. There may be circumstances in which certain users do not need full access to all the information on a policy. For example, a user is linked to a policy through a producer code with a status of **Suspended** or **Terminating**. You may want to limit the actions that user can take on that policy. Similarly, a user linked to a policy through the producer code of record but not the producer code of service only needs to see basic information related to their commission. Security can be fine-tuned in these ways by using permission handlers, defined in the `security-config.xml` file, to wrap system permissions to create application permissions. The permission handler can be used to define how the system permission interacts with the producer code status and producer code of record. The application permissions can then be used in PCF and Gosu to restrict access to system features, rather than using the underlying system permission.

## Security Restrictions Using the Status Field

In previous sections, you learned that producer codes place restrictions on which policies and accounts you can access and which actions you can perform. Therefore, producer codes provide an additional level of security in PolicyCenter. However, through producer code security, restrictions can be placed upon access beyond the permissions associated with the matching producer code. Both producer code and organization contain a status field, which uses a typelist. To see this, select **Administration** → **Users & Security** → **Producer Codes**. In the **Producer Codes** screen, select a **Status** from the drop-down typelist.

You can use status to limit access to an agency whose contract has expired or for a producer code that has been misused. The default statuses are:

- Active
- Limited
- Suspended
- Terminating
- Terminated

Both the status of the matching producer code and the status of the user's organization must be checked before allowing access through producer code security. Both status fields must either be **Active** or on the list of allowed `status_values` in the permission handler. Without any configuration to the security handlers, full permissions are granted when the status is **Active**. No permissions are allowed if the status is anything else, except for renewals, which allow for **Limited** status. This behavior allows a producer to continue maintaining his business through renewals, but does not allow him to write new business. In the default configuration, there is not an automated mechanism to update the producer code and organization status fields. An administrator must update the status fields at the appropriate time.

## Producer of Record and Producer of Service

PolicyCenter tracks two types of producers, each represented by a producer code. The two types of producers are:

- **The producer of record** brings the business to the carrier, and therefore receives the commission based upon the premium. However, that producer may not be servicing the insured, usually because the insured requested a change to another producer. A producer of record, if not actively servicing the account, may have a more limited set of permissions on the policy and account.
- **The producer of service** is the person who is actually servicing the policy. The producer of service has the permissions to edit policies, and, at the time of renewal, becomes the producer of record. At that point, the agent is eligible to receive commissions. At any given time, there is a single producer for a policy who assumes full rights on the policy.

### Producers of Service Can Edit the Account

The producer codes that grant access to an account are the producer codes of service for policies on that account, plus the producer code originally used to create the account. The producer codes associated with unbound submissions are not included on the list of account producer codes.

On a new account, the producer code associated with the first submission is typically the same producer code used to create the account. If the producer code on a non-issued submission changes, then:

- The new producer code becomes the producer code of record and the producer code of service on the submission.
- The new producer can edit the submission, but not the account.
- The old producer code is not automatically deleted from the account. However, since there is no longer a relationship to the submission, the old producer cannot view or edit that submission.

When the policy is issued, the new producer becomes the producer of record and the producer of service for that policy. That producer code is then added to the list of producer codes for the account.

### Changing the Producer

After a policy has been issued, you may need to change the producer through a policy change, but that change does not amend the producer code of record. Instead, PolicyCenter sets a producer code of service, separate from the producer code of record. However, when a policy is renewed or rewritten, the renewal or rewrite automatically sets the producer code of record to the servicing producer code. However, if you explicitly change the producer code during the renewal process but before the policy is bound, then that change affects the assignment of the producer code of record.

In PolicyCenter, you can edit the producer code of record in a submission, issuance, rewrite, or renewal job. The producer code of record is implicitly also the producer code of service for the policy. At any given time, there is only one producer code of service for the policy, though each contractual period\_might have different producer codes of record. The producer code of service is the only producer that has full permissions on the policy. A policy change allows editing of the producer code of service only, not the producer code of record.

### Adding a Third Producer

Assume a policy has a separate producer code of record and producer code of service. What if the user makes another mid-term change and sets the producer code of service to a third producer? Upon binding the policy change, the third producer is added to the account and can edit the account and policy. The second producer, who had previously been the producer of service, can no longer edit the policy. The second producer is removed from the account, so long as that producer code is not the producer code of service for any other policies on the account.

## Managing the PolicyCenter Community

Users represent individuals who are granted access to PolicyCenter. You can view and manage user information in the **Administration** tab. Use the links in the sidebar to:

- Search or edit users, groups, organizations, regions, or producer codes.
- View, edit, or create roles or authority profiles.
- Define attributes that the system can use for making assignment decisions.

Typically, the user belongs to at least one group that represents the actual reporting relationship found in the company's organizational chart. This group is the main group that the user works in and the supervisor that the user normally reports to. Users can belong to an unlimited number of groups or need not belong to any group. If you define groups based on relationships other than organization, your users may end up belonging to multiple groups. For example, a user might be a part of the **West Coast Division** and a member of the company wide group of **System Administrators**.

When you first install PolicyCenter, you may find it more convenient to use the `import_tools` command to import user information – especially if you have a large number of users. See “Importing and Exporting Administrative Data” on page 81 in the *System Administration Guide* for more information.

### Understanding Internal and External Administration

PolicyCenter administration allows you to manage the information access of both internal employees and external agents. External agents are people that act as intermediaries between clients and insurers that offer policies. You control the access of both internal and external employees by using roles and producer codes. You define the permissions associated with each role and assign them to users.

Within PolicyCenter, an external company of producers, such as an agency or brokerage, is known as an *organization*. You can categorize the employees within an external organization or within your own employer, the insurer, by creating Group objects. Defining the structure of your company's groups depends on your business requirements. Typically, administrators create groups according to such things as geographical regions, divisions, and departments. You can also categorize people across these boundaries by other attributes such as job classification, for example, all system administrators.

Because managing both the internal company and the external company can become quite complex, in PolicyCenter you can delegate management of external companies to a designated external **User Admin**. Do this by assigning the **User Admin** role to a user within that external organization. That user will have permission to administer users and groups within the user's organization, but will not have access to other organizations. Delegating allows a **User Admin** to build the set of users and groups that represent an external company. Even though you can delegate administration of these external companies, you still have access and complete administrative rights from the **Administration** tab.

### Creating External Users

External users represent people outside your company who may need access to PolicyCenter data. Producers are typically external users who perform tasks such as submitting a policy. External users can also represent people who must be known by the system for assignment purposes, even if they never actually log in.

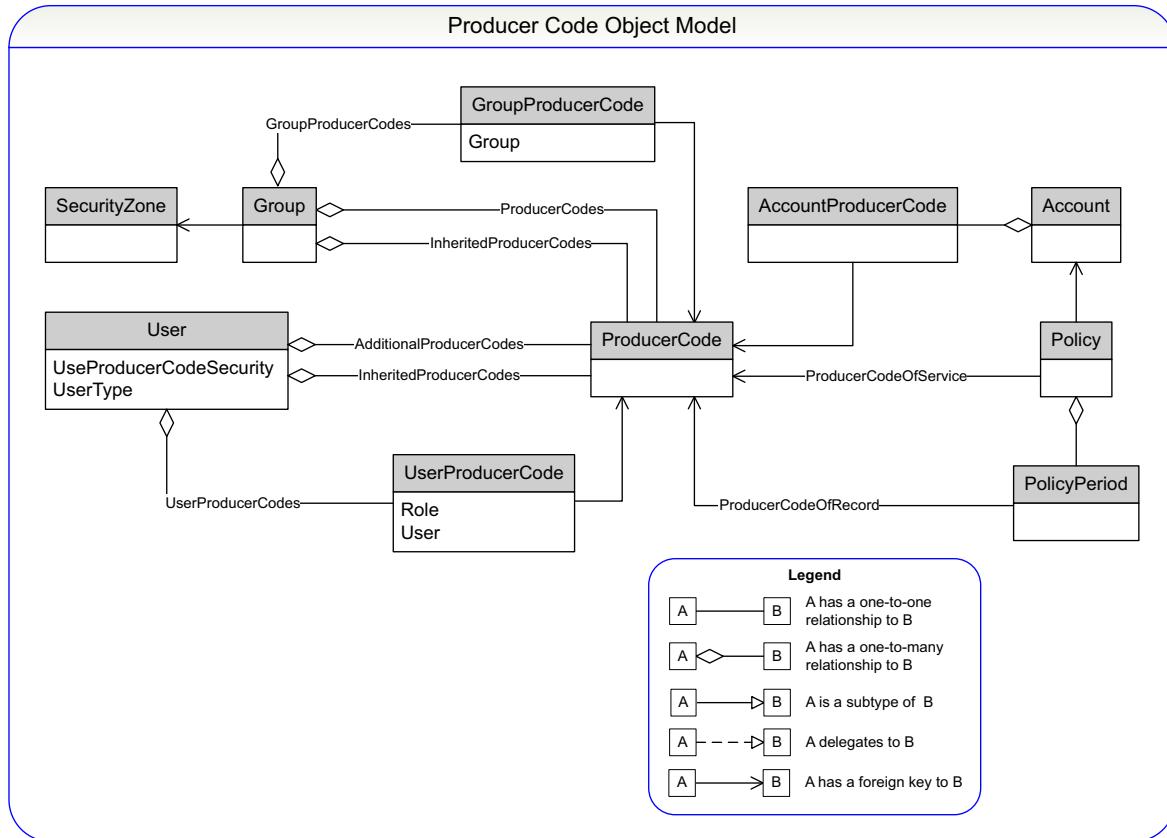
To create an external user, set the **Internal User** option set to **No** on the **Basic** tab in the **User** screen. When you create an external user, the fields and available values in the **User** tabs change. Required fields appear with an asterisk: \*. Unlike internal users, external users must be explicitly associated with an **Organization**. Internal users are implicitly associated with the insurer organization. The tree browser as seen on your **Administration** tab always shows the user's home organization (either internal or external). If the user is internal, the tree also shows the last external organization viewed.

# Security Object Models

This topic contains object model diagrams for PolicyCenter security.

## Object Model for Producer Codes

The following illustration shows some of the key relationships of the ProducerCode entity.



The GroupProducerCodes array contains the GroupProducerCode entities available to the Group. The GroupProducerCode entity has foreign keys to the ProducerCode and Group entities. The Group entity has two derived arrays of ProducerCode entities. These contain the producer codes which apply to the users in the group. The ProducerCodes derived array contains the list of producer codes associated with this group. The InheritedProducerCodes derived array contains the producer codes inherited from parent groups. (Remember that groups can be arranged in a hierarchy of groups.)

The UserProducerCodes array contains the UserProducerCode entities available to the User. These are the producer codes assigned directly to the user. The UserProducerCode entity has foreign keys to the ProducerCode, Role, and User entities. The AdditionalProducerCodes derived array provides another way to access the producer codes directly assigned to the user.

The User entity has an additional derived arrays of ProducerCode entities. The InheritedProducerCodes derived array contains the producer codes inherited from the group or groups that the user belongs to. These producer codes apply to the user.

# Working with Security

To control the level of access to PolicyCenter data, you manage it primarily from the **Administration** tab in the user interface, and to a lesser extent, in plugins.

This topic includes:

- “Viewing Permissions on Selected Roles” on page 647
- “Creating and Removing a Permission” on page 647
- “Adding a Permission to a Role” on page 647
- “Adding or Removing a Role” on page 648
- “Turning On Producer Code Security” on page 648
- “Designating a Client Data Integration Handler” on page 648

## Viewing Permissions on Selected Roles

### To see or edit the list of permissions on selected roles

1. Navigate to the **Administration** tab. From the sidebar, click **Roles**.
2. Select the link of the named role.

## Creating and Removing a Permission

### To create a new permission

1. In Studio, navigate to **TypeLists** → **SystemPermissionType**.
2. Click **Add**. Enter the code, name, and description.
3. Certain PCF files (screens) may be visible if the user has this permission. Add code in the **editable** advanced property of the PCF file which determines whether the screen is visible. For example, set the **editable** advanced property to a permission so that it is true when a user has that permission:  
`perm.System.editsubmission`

### To remove a permission

1. In Studio, navigate to **TypeLists** → **SystemPermissionType**.
2. Select the permission, and click **Remove**.
3. Remove all references to the permission in PCF, Gosu, and other files in the application.

**Note:** Deleting permissions from an existing role is not a good idea, since users who needed the deleted permissions will be adversely affected. Instead, create a new role without that permission, and assign the new role, rather than the old role, to new users.

## Adding a Permission to a Role

### To add a permission to a role

1. Select **Administration** → **Users & Security** → **Roles**.
2. Select the role that you would like to add the permission to. PolicyCenter displays the **Role** screen.
3. Click **Edit**, then click **Add** under **Permissions**.
4. Select the permission from the list.

5. Click **Update** to save your changes.

**Note:** You create and modify roles, and assign roles to (or remove roles from) users from the **Administration** tab in the user interface.

## Adding or Removing a Role

This topic describes how to add and remove a role. You can select whether a role is internal or not. You cannot assign internal roles to external users.

### To add a new role

1. Select **Administration** → **Users & Security** → **Roles**.
2. On the **Roles** screen, click **New Role**.
3. Enter the **Name**, select a **Type**, indicate whether it is an internal role or not, and optionally give a description.
4. Click **Add**, and then select the permissions.
5. Click **Update**.

### To Remove a Role

1. Select **Administration** → **Users & Security** → **Roles**.
2. On the **Roles** screen, select the roles you wish to remove.

**Note:** Removing a role is often not a good idea, especially for those users who will lose permissions they previously had.

## Turning On Producer Code Security

Users for whom producer code security is turned on can only use their role permissions against policies that have a producer code matching one of their own producer codes.

### To turn on producer code security

1. Select **Administration** → **Users & Security** → **Users** and navigate to a user.
2. In the **User** screen, select the **Access** tab.
3. In **Use Producer Code** security, select **Yes**.

### See also

- “Producer Code Security and Policies” on page 639

## Designating a Client Data Integration Handler

Sometimes ContactManager cannot create new contacts that PolicyCenter sends to it. Examples of problems that prevent creation include bad IDs, incomplete information, and duplicate contacts. Whenever ContactManager fails to create a new contact, ContactManager notifies PolicyCenter, which assigns an activity to a user whom you designate to follow up and resolve such issues.

The user who receives these activities from PolicyCenter must have permissions that enable working with contacts. The following table shows the set of permissions that are required in the base configuration.

Permission name	Permission code
Create address book contact	abcreate
Create contact with any tag	anytagcreate

Permission name	Permission code
Create local contacts	ctccreate
Delete address book contacts	abdelete
Delete contact with any tag	anytagdelete
Delete local contacts	ctcdelete
Edit address book contacts	abedit
Edit contact with any tag	anytagedit
Edit local contacts	ctcedit
View address book contacts	abview
View address book contact search pages	abviewsearch
View contact with any tag	anytagview
View local contacts	ctcview

These privileges grant the designated user access to local contacts and to contacts in ContactManager. To ease administration for changes to the designated user, create a user role, such as Client Data Integration Handler. Then, assign your designated user to that role. Assign only one user to that role at any time.

Before PolicyCenter can assign activities to your designated integration handler, you must configure the PolicyCenter method `ABContactSystemPlugin.getAdminUserForIntegrationHandling` to use the designated user's PolicyCenter login name. If you change the designated user in PolicyCenter, manually reconfigure this method with the newly designated user's login name.

#### See also

- To learn how to configure `ABContactSystemPlugin.getAdminUserForIntegrationHandling` with a login name, see “Step 1: Integrate ContactManager with PolicyCenter” on page 51 in the *Contact Management Guide*.

## Security and Configuration Scenarios Related to Producer Codes

This topic describes common security scenarios related to producer codes that require configuration.

### Producer Codes Assigned by Level

One of the most common producer code models employed by carriers is that agents are given different levels that define their rights within the system. For example, a Level 1 Producer has rights to enter submissions, but nothing else. A Level 2 Producer can execute submissions, policy changes, and renewals. A Level 3 Producer can execute submissions, policy changes, renewals, cancellations, reinstatements, and rewrites. Producer codes are assigned on an individual basis. Each user has just one producer code that they use for all their business.

Under this model, there would be three roles of type User Producer Code:

- Producer Level 1
- Producer Level 2
- Producer Level 3

There would also be a separate producer code defined for each agent. An agent who is Level 1 would have the Producer Level 1 role assigned to both the User and the Producer Code. The set of permissions in Producer Level 1 defines everything about that agent's access to PolicyCenter. Since there is one producer code for each agent, there is no need to assign producer codes to groups at all, they can be assigned explicitly to the users.

Also, you can use the permission handlers to limit access for producers of record and for producers who have a suspended or terminated status.

## Producer Codes Assigned by Line of Business

Another common producer code model is that a group of producer codes is given to the agency, one for each line of business (LOB). All agents in the agency use the same producer codes, but not all agents necessarily work with all LOBs. For example, one set of agents handles personal lines exclusively and another set handles only commercial lines.

Under this model, the agency organization would be structured with two groups:

- Personal
- Commercial

Since the users and producer codes each apply to only personal or commercial, they are each assigned to only one of the two groups. Users inherit the producer codes through their group, and do not need to have producer codes assigned explicitly.

Within commercial lines, you may not want to assign the same set of rights for all lines of businesses. For example, for business auto, you may want agents only to do submission, policy change, and renewal jobs. However, for workers' compensation, agents can do only submissions and policy changes.

To assign different rights to different lines of business, create a single user role, Producer, and assign it to all agents. This role could contain the superset of permissions that an agent is expected to be able to use. This role includes permissions for submissions, policy changes, and renewals. Then create separate producer code roles for each job type:

- Producer: Submission – Permissions such as `createsubmission` and `editsubmission`
- Producer: Policy Change – Permissions such as `createpolchange` and `editrenewal`
- Producer: Renewal – Permissions such as `createrenewal` and `editrenewal`
- Producer: All – General permissions that apply regardless of job type, such as `viewpolicyfile`

The producer codes for business auto would then be given the following roles:

- Producer: All
- Producer: Submission
- Producer: Policy Change
- Producer: Renewal

The producer codes for workers' compensation would only be given the following roles:

- Producer: All
- Producer: Submission
- Producer: Policy Change

By using this model, there would also be a configuration in the `performNameClearance` method in the `AccountPlugin`. You must add code to prevent each producer code from being used on an inappropriate line of business. Finally, use the permission handlers to limit access for producers of record and for producers who have a suspended or terminated status.

## Producer Codes Assigned by Level and Line of Business

You can define a producer code model with the producer codes assigned to an agency by line of business, but each agent still has a level. (The previous section describes how to assign producer codes by line of business.) There would be multiple user roles, each representing a different level of producer. There would be producer code roles as well, each with a different slice of overall producer functionality, such as by job type. An agent's level, as defined by their user role, limits what individual agents can do on any policy. Producer code roles further limit what agents can do. The carrier assigns the appropriate producer code roles to the producer codes.

## Producer Codes Roles Customized by User

PolicyCenter supports an even deeper level of security configuration by user. Building upon the previous example, suppose that there is an agent who is generally Level 3, with broad rights within PolicyCenter. Other agents within the agency are also Level 3. The agency has producer codes that multiple agents use. Each producer code is associated with a variety of different permissions. The agent is generally Level 3, and Level 3 generally has the rights to perform submissions, policy changes, and renewals. However, this agent has not yet been properly trained on the renewal process for workers' compensation. You can configure the system to prohibit this agent from doing renewals.

### Two Possible Ways to Configure

There are two ways to prohibit an agent from doing renewals. The first way is to create a new application permission, specific to workers' compensation renewals. This application permission is checked in the appropriate PCF files for workers' compensation renewals. There would be a separate user role for Producer - Level 3 without WC Renewal that does not have the WC renewal permission. However, if this type of situation is common, creating many new permissions would make maintenance of the carrier's configuration challenging.

The second possibility is to use a deeper level of security configuration. You can configure the producer code roles allowed for a particular producer code by a particular user. By default, all producer code roles associated with the producer code are allowed for users given that producer code. But an administrator can turn off specific roles for just that user on just that producer code. In this use case, the administrator unchecks the Producer - Renewal role for the workers' compensation producer code for this agent in the **Administration** tab. The administrator leaves the other producer code roles enabled. Therefore, just that user's permissions are limited more for just that one producer code.

## Producer Code Currency

On the **Administration → Users & Security → Producer Codes** screen, each **Producer Code** has a **Currency** field if multicurrency display is enabled. Use this field to specify the settlement currency in which the producer can bind policies.

Although the user interface in the base configuration enables you to select only a single currency for each producer code, the data model supports multiple currencies. The producer code entity (**ProducerCode**) has an array of currencies (**ProducerCodeCurrencies**).

## Security Dictionary

The PolicyCenter *Security Dictionary* is web-based documentation that you can generate as part of the PolicyCenter installation. Every time that you change the datamodel, you must regenerate this dictionary.

Use the *Security Dictionary* to view:

- **Application permission keys** – You can view entity keys individually or grouped by entity. Click the **Summary** link to view the permission keys by entity.
- **Pages** – Select a page to see which permissions that page uses. These correspond to PCF files.
- **System permissions** – Select a permission to see any associated roles, related application permission keys, related pages, and related elements. For example, if you select **completeaudit**, which is the permission to complete an audit, then you see that the Audit Examiner, Audit Supervisor, and Superuser roles use this permission. The related application permission keys are **Audit complete** and **PolicyPeriod audit**. Knowing which PCF files contain this permission is also useful for troubleshooting as you can see if the permission is used correctly on those pages.
- **Roles** – Roles contain the same information that you can see by selecting **Administration → Users & Security → Roles**. The value of seeing it from the *Security Dictionary* is that you can see which other roles share that

permission. For example, if you select **Producer**, you see the list of permissions that a producer has. If you select a permission such as **createsubmission** (which is the permission to create a submission), then you also see which roles share that permission. In this example, Producer Code - Submissions, Superuser, Underwriter, Underwriter Assistant, and Underwriting Supervisor can create submissions, too.

**See also**

- “Regenerating the Data Dictionary and Security Dictionary” on page 30 in the *Configuration Guide*

## Access Control for Documents and Notes

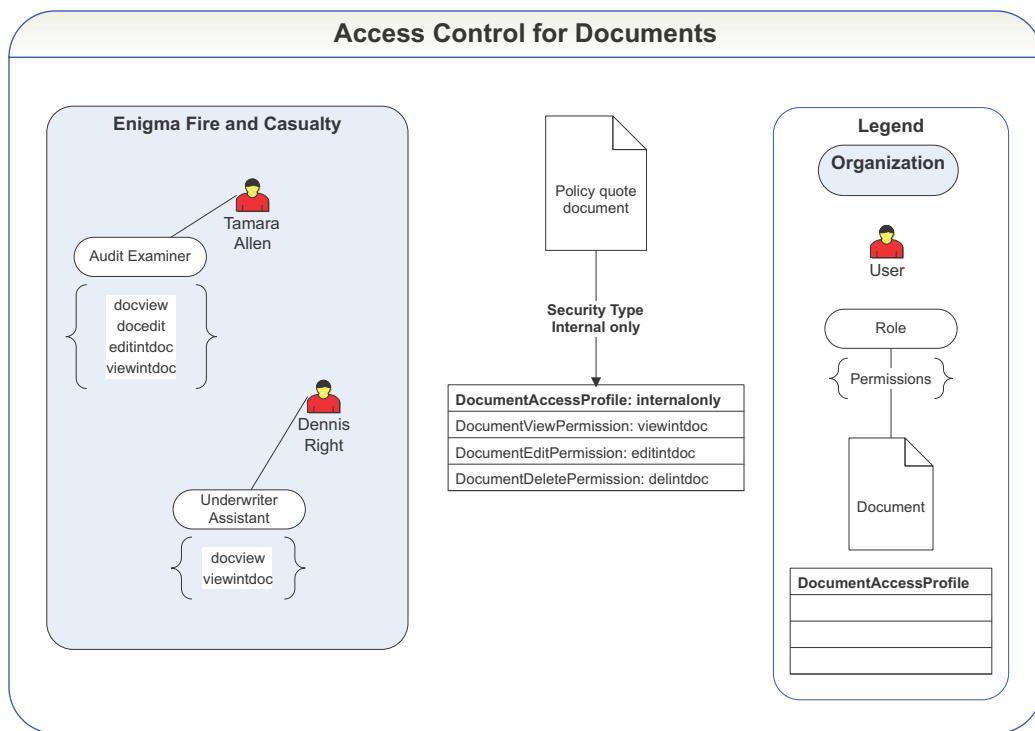
In addition to the standard document and note-related system permissions, you can control access to documents and notes by configuring access permissions. To do so, a document must have its document security type set. To see documents of a particular type, you must have both permission to view documents in general and permissions to access to the document security type. A document access profile grants this access. Access control for notes is the same as for documents.

Note and document access control requires:

- **Document Security Types or Note Security Types** – These are document subsets (like **internalonly**, **sensitive**, or **unrestricted**) for access control. These security types are defined in the **DocumentSecurityType** typelist. They appear in the **Security Type** dropdown of the **Documents → Document Details** and **New Document** screens. You can assign a document at most one of these security types. Similarly, the **NoteSecurityType** typelist defines the types of note security available.
- **System Permissions** – Users must be assigned roles containing permissions to access documents and notes in general. The roles must also have permissions which match those in the access profile of the security type. Different permissions affect notes and documents.
- **Document and Note Access Profiles** – Using the above two concepts, these profiles relate permissions and security types to restrict access to a subtype of documents.

In the illustration below, the policy quote document has **Security Type** set to **Internal only**. Tamara Allen is an audit examiner. Because she has the **Audit Examiner** role, she has the necessary permissions to view and edit **Internal only** documents. Therefore, she can edit and view the policy quote document.

Dennis Right is an underwriter assistant. His permissions only allow him to view **Internal only** documents.



## Working with Access Control for Documents and Notes

The following sections describe how to define and use document and note access control.

- “To Create Document and Note Security Types” on page 653
- “To Assign a Document or Note to a Security Type” on page 653
- “To Create Document Access Profiles and Note Access Profiles” on page 653

### To Create Document and Note Security Types

A document type is set using the **Security Type** field in the user interface or through Gosu. The **DocumentSecurityType** typelist contains the **internalonly**, **sensitive**, and **unrestricted** security types. You can add your own security types. Documents which are not assigned a special security type are given the **unrestricted** security type.

### To Assign a Document or Note to a Security Type

After creating a new document by selecting **Actions** → **New Document**, choose the security type to assign the document from the **Security Type** dropdown list.

### To Create Document Access Profiles and Note Access Profiles

Access to document types is controlled by adding a document access profile section to **security-config.xml**. You must have a document access profile for each document security type you want to place under document access control. The same is true for notes.

Each document access profile has this syntax, where **type** specifies a document or note security type, and **perm** is a system permission:

```
<DocumentPermissions>
 <DocumentAccessProfile securitylevel="type">
 <DocumentViewPermission permission="perm" /> <!-- define for each security type -->
 <!-- allow this permission to view-->
```

```
<DocumentEditPermission permission="perm"/> <!-- allow this permission to edit-->
<DocumentDeletePermission permission="perm"/> <!-- allow this permission to delete-->
</DocumentAccessProfile>
</DocumentPermissions>

...
<NotePermissions>
 <NoteAccessProfile securitylevel="type"> <!-- define for each security type -->
 <NoteViewPermission permission="perm"/> <!-- allow this permission to view-->
 <NoteEditPermission permission="perm"/> <!-- allow this permission to edit-->
 <NoteDeletePermission permission="perm"/> <!-- allow this permission to delete-->
 </NoteAccessProfile>
</NotePermissions>
```

In the default configuration, PolicyCenter provides these document permissions:

```
<DocumentPermissions>
 <DocumentAccessProfile securitylevel="unrestricted"/>
 <DocumentAccessProfile securitylevel="internalonly">
 <DocumentViewPermission permission="viewintdoc"/>
 <DocumentEditPermission permission="editintdoc"/>
 <DocumentDeletePermission permission="delintdoc"/>
 </DocumentAccessProfile>
 <DocumentAccessProfile securitylevel="sensitive">
 <DocumentViewPermission permission="viewsensdoc"/>
 <DocumentEditPermission permission="editsensdoc"/>
 <DocumentDeletePermission permission="delsensdoc"/>
 </DocumentAccessProfile>
</DocumentPermissions>
```

There is a similar set of permissions for notes in the `NotePermissions` element.

# Authority Profiles

*Authority profiles* determine the types of underwriting issues that a user can approve. You can assign authority profiles to users. Each authority profile contains one or more *authority grants* which grant levels of approval, often for a specific underwriting issue and within certain limits. Authority grants are similar to physical letters of authority.

This topic includes:

- “Authority Profile Overview” on page 655
- “Working with Authority Profiles” on page 657

**See also**

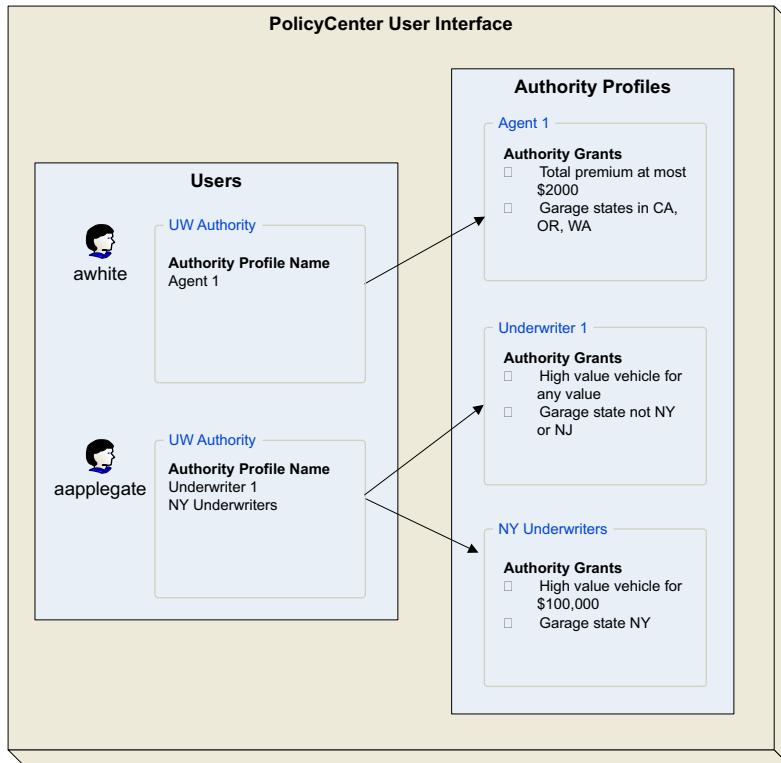
- “Underwriting Authority” on page 411
- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*

## Authority Profile Overview

Authority profiles determine the types of underwriting issues that a user can approve. You can assign authority profiles to users. You can assign each user to one or more authority profiles. The authority profile consists of authority grants. Each authority grant is based on an issue type defined in the UWIssueType system table. The authority grant can specify the values which need additional approval.

The following diagram shows two users with different authority profiles. User awhite has an Agent 1 authority profile. User aapple aggregate has two authority profiles which each contain a different default value for the high-value vehicle grant. With authority grants that have numeric comparators, the grant with more risk takes priority. Therefore, aapple aggregate has the authority to approve a high value vehicle of any value. This user also has an authority grant for garage jurisdiction not NY or NJ and another for garage jurisdiction NY. With authority

grants for sets, the grants are additive. Therefore, aapplegate can approve vehicles for all garage jurisdictions except NJ.



UWIssueType system table			
Code	Comparator	BlockingPoint	CheckingPoint
PA: Total premium	At most	Blocks Quote Release	Created Before Quote
PA: Garage States	In set	Blocks Quote Release	Created Before Quote
PA: HighValueAuto	At most	Blocks Bind	Created Before Quote

The base application defines the following authority profiles:

- Agent 1
- Agent 2
- Underwriter 1
- Underwriter 2
- Underwriting Manager

You define authority profiles in the **Administration** tab of PolicyCenter. You define authority grants in the **UWIssueType** system table.

### Agent 1 Example

In the base configuration, the Agent 1 profile has the following authority grants for personal auto policies:

- The collision deductible is at least \$1000. A user with the Agent 1 profile cannot approve a deductible less than this.
- The comprehensive deductible must be at least \$500. A user with the Agent 1 profile cannot approve a deductible less than this.
- Can approve the **Garage State** underwriting issue if the jurisdiction of garaging is California, Oregon, or Washington. Any other jurisdiction triggers a Garage states underwriting issue.

- The total premium can be \$200 at most. A user with the Agent 1 profile cannot approve a total premium greater than this amount.

## Working with Authority Profiles

Authority profiles define levels of underwriting authority through authority grants. You add authority profiles to users.

### Viewing or Editing an Authority Profile

Follow these steps to view or edit an authority profile.

1. Click the **Administration** tab.
2. Click **Authority Profiles** in the left sidebar. PolicyCenter displays the list of authority profiles.
3. Click an authority profile such as **Agent 1**. PolicyCenter displays the authority grants for Agent 1.  
Or, click **New** to add a new authority profile.
4. Click **Edit** to add or modify an authority grant.
5. Click **Add** to add an authority grant. Select a **Type**.

The authority grant is defined in the **UWIssueType** system table. For more information about this table, see “Underwriting Issue Type System Table” on page 467 in the *Configuration Guide*. The value in the second column is defined in this table. For **At most**, **At least**, and **In Set** grants, you can specify the amount or set in the third column. The user with this authority grant can approve amounts or jurisdictions within these limits. The authority grant can have a default offset or percentage specified in the **UWIssueType** system table. If so, the approval limit is increased or decreased by this amount.

### Assigning an Authority Profile to a User

The **User** screen has a **UW Authority** tab. You can add one or more authority profiles to a user in the **UW Authority** tab.



# Team Management

PolicyCenter provides a management tool that helps supervisors and managers manage their groups. This tool displays the number of activity and policy transaction instances grouped by time and status. You access this tool from the **Team** tab, and use it to control and distribute the activity and workload of your team. This topic describes the team management tool.

This topic includes:

- “Team Management Overview” on page 659
- “Working with the Team Tab” on page 662

**See also**

- “Configuring the Team Tab” on page 489 in the *Configuration Guide*

## Team Management Overview

Supervisors and managers can manage their teams, obtain status information, monitor loads, identify backlogs, and reassign activities by using the team management functionality in PolicyCenter. You can access this functionality from the **Team** tab. In some respects, team management is a reporting tool in which you can see workload summaries by group. You can then navigate to view and manage the workload of a team member. The reporting categories for workloads are: **Summary**, **Activities**, **Submissions**, **Renewals**, **Other Policy Transactions**, and **Misassigned**. This topic also provides information about reporting categories.

When you first select Team, PolicyCenter displays to **My Groups** summary as shown in the following illustration for the svisor user.

Name	Open	Overdue	Completed	Open	New	Bound	Open
Actuary Unit	0	0	0	0	0	0	0
Eastern Mid-West Regi...	0	0	0	0	0	0	0
Eastern Region	2	0	0	0	0	0	0
Finance & Treasury	0	0	0	0	0	0	0

1. Your groups. Select a group to display information for that group or user.
2. Links to reporting categories.
3. Displays the time that this information was last calculated.

## Groups and the Team Tab

PolicyCenter organizes users into groups and subgroups. The **Team** tab displays a root group (**My Groups**) that contains other subgroups (**Eastern Region Underwriters** and **Los Angeles Branch UW**). Each group contains individual users, queues (**In Queue**), and misassigned (**Misassigned**). **In Queue** contains activities that have not been assigned to anyone. The **Misassigned** category contains policy transactions and activities assigned to the group under which the node appears, but the policy transaction or activity was assigned to an invalid user. An invalid user is usually someone who is no longer a member of the group. The user may have switched groups or left the company.

Selecting a node in the tree hierarchy on the left side of the screen updates the main work area with information for that group or user. Select a reporting category to see information for that group or user.

## Team Tab User Categories

The **Team** screens show the policy transaction workload for groups or individual users. The **Team** screen determines whether a user has work to do on a policy transaction depending upon whether the user is a *by-role* or *by-activity* user.

- **By-role** – These users work on policy transactions when they have a participating role. A customer service representative (CSR) is an example of a by-role user. The CSR is the producer for a policy transaction and makes sure that the policy transaction progresses. The CSR may not have any activities associated with the policy transaction. For this category of user, the **Team** screen displays policy transactions on which the user has a role.
- **By-activity** – For certain types of users, having a role on a policy transaction does not represent work. An underwriter may have a role on thousands of policy transactions. For example, one underwriter may have a role on the large number of policies generated by a group CSRs. For a personal lines which renew every six

months, a third or more of the policies may have an active renewal in process at any time. One underwriter has a role for each renewal even if the renewal is expected to complete automatically without intervention. Even though the underwriter has a role, these renewal policy transactions do not represent work for the underwriter unless a policy transaction requires some action. An activity makes the underwriter aware that work is required on the policy transaction. For this category of user, the **Team** screen displays policy transactions on which the user has an activity.

For by-role users, the detail view lists all policy transactions on which the user has a role. For by-activity users, the detail view lists only policy transactions on which the user has an activity. In both cases, the list displays information based on the category of the user being viewed, not the user viewing that information.

Renewal policy transactions are an exception. The **Team** screen displays renewal policy transactions by-activity for both categories of users.

The **Summary** screen displays both by-activity and by-role policy transactions and policy transactions. You can select **View Policy Transaction** to view policy transactions **By Role** or **By Activity**. The default view is **By Role**. The **View policy transaction** selector does not appear for **In Queue** or an individual user because the user type determines the **Team** screen display.

In the base configuration, the **Team** tab determines the user category by checking the **User.UserType** property. In the base configuration, the **UserType** typecodes are mapped to **Team** tab user types as follows:

UserType typecode	Team tab user category
assistant	By-activity
auditor	By-activity
other	By-activity
producer	By-role
underwriter	By-activity
unknown	By-activity

## Reporting Categories

The reporting categories provided with PolicyCenter are:

- **Summary** – View summary statistics of each group’s activities, submissions, renewals, and other policy transactions.
- **Activities** – View activities for a group or user.
- **Submissions** – View submissions for a group or user.
- **Renewals** – View renewals for a group or user.
- **Other Policy Transactions** – View other works orders, including policy changes, cancellations, reinstatements, renewals, and audits, for a group or user.

The summary reporting category displays totals calculated for each policy transaction and activity for each user in each group. Totals are also computed for invalid user-group pairs where the user has changed groups since the creation and assignment of that policy transaction or activity. Finally, activity count calculations are performed for queues belonging to a group but nothing else since the others cannot be assigned to a queue.

For activities, you can assign one or more items to another user, group, or queue. For policy transactions, you can assign a policy transaction role to a user on one or more items. You cannot reassign the Creator policy transaction role.

### See also

- “Assigning Activities” on page 662
- “Assigning Submissions, Renewals, and Other Policy Transactions” on page 662

# Working with the Team Tab

This topic describes how to work with the **Team** tab in the PolicyCenter user interface.

## Assigning Activities

The **Activities** reporting category screen displays the activities for the current selection in **My Groups** in the left sidebar. You can assign one or more activities to a user, group, or queue.

### To assign one or more activities

1. Select the **Team** tab.
2. Make a selection in the left sidebar. To view activities for all of your groups, select **My Groups** at the top.
3. Select the **Activities** reporting category.

PolicyCenter displays the activities for the current selection in the left sidebar. A drop-down list lets you choose to view **Open**, **Overdue**, and **Completed** activities. A check box appears to the left of each activity.

4. Select one or more activities and click **Assign**.

PolicyCenter displays the **Assign Activities** screen. You can do the assignment in one of the following ways:

- **Select from list** – Select from a drop-down list. PolicyCenter lists the groups that the current user is in. The drop-down list also contains potential assignees based upon the users involved in the policy and other members of the groups associated with the user or the policy.
- **Find a user, group, or queue** – Search for a user, group, or queue.

5. Make a selection, and click **Assign**.

## Assigning Submissions, Renewals, and Other Policy Transactions

The **Submissions**, **Renewals**, and **Other Policy Transactions** reporting categories screens display the policy transactions for the current selection in **My Groups** in the left sidebar. For the selected policy transactions, you can choose a user for an assignment role such as auditor, producer, or underwriter. The selected user replaces the user who previously held that assignment role. If a user is a member of more than one group, you must also assign the group.

**Note:** This functionality is also available in the **Search Results** on the **Search Policies** screen. For more information, see “Advanced Search for Policies” on page 64 and “Working with the Advanced Search Tab” on page 65.

### To assign a user to one or more policy transactions

1. Select the **Team** tab.
2. Make a selection in the left sidebar. To view policy transactions for all of your groups, select **My Groups** at the top.
3. Select the **Submissions**, **Renewals**, or **Other Policy Transactions** reporting category.

PolicyCenter displays the policy transactions for the current selection in the left sidebar. A drop-down list lets you filter the policy transactions as follows:

- **Submissions** – **Open**, **New**, and **Bound** policy transactions.
- **Renewals** – **Open**, **New**, **Renewed**, **Non-Renewed**, and **Not Taken** policy transactions.
- **Other Policy Transactions** – **Open**, **New**, and **Approved** policy transactions.

A check box appears to the left of each activity.

4. Select one or more policy transactions and select a role from the **Assign** drop-down list.

**Note:** The **Creator** role is not in the drop-down list because you cannot reassign this role.

PolicyCenter displays the **Assign policy transactions** screen. For each policy transaction, this screen displays the type, policy transaction or policy number, and role.

5. Enter a **User Name, First Name, or Last Name** and click **Search**.

PolicyCenter displays matching users. For each user, PolicyCenter displays the **Group** and **Parent Group**. A user appears multiple times if the user belongs to more than one group. For example, you search for users with last name Applegate. The search returns two rows for Alice Applegate because Alice is a member of the Eastern Region Underwriting and Los Angeles Branch UW groups.

6. In the search results, click **Assign** to assign a user for the chosen role. If a user is a member of more than one group, select the user's row that displays the group of your choice.



# Policy Holds Administration

Policy holds allow a carrier to prevent users from creating new policies or changing existing policies in a specific region for a period of time. PolicyCenter provides policy holds for the following reasons:

- Natural disaster – The carrier can put a hold on issuing or changing policies in a region affected by a natural disaster.
- Regulatory changes – The carrier can put a hold on issuing or changing policies during a period of time when the carrier is finalizing insurance rates or changing coverage forms.

This topic includes:

- “Policy Holds Overview” on page 665
- “Working with Policy Holds” on page 668
- “Policy Hold Screen” on page 672
- “Policy Hold Actions in Renewals” on page 673
- “Policy Hold Object Model” on page 674
- “Configuring Policy Holds” on page 674
- “Importing and Exporting Policy Holds” on page 677

## Policy Holds Overview

PolicyCenter provides policy holds for natural disasters and for regulatory changes. Policy holds are types of underwriting issues.

**See also**

- “Underwriting Authority” on page 411

### Underwriting Holds for Natural Disasters

PolicyCenter provides underwriting policy holds for natural disasters. You can base the policy hold on written date or effective date.

When a natural disaster is impending or in progress, a carrier may decide not to bind any new policy transactions in that region before or after the natural disaster. The carrier does not want to write new policies until the natural disaster passes and the losses are assessed. As the natural disaster progresses and the potential losses change, the carrier may need to modify the underwriting hold.

For example, the weather service predicts that a hurricane will hit the state of Hawaii. The state of Hawaii consists of five counties. The weather forecast says that the hurricane threatens only Kauai County. Customers start calling in to add flood coverage to their policy. The carrier puts an underwriting policy hold on Kauai County so that agents cannot add that coverage. The carrier does not set a hold end date because the storm's progress and amount of time to assess damages are unknown.

On the second day, the storm has not yet hit Kauai County. However, the hurricane now also threatens Honolulu County and part of Maui County. The carrier adds those counties to the underwriting hold. Because the storm has not yet arrived, the carrier updates the hold start date to the current date.

On the third day, the hurricane bypasses Kauai County but still threatens Honolulu County and part of Maui County. The carrier removes Kauai County from the policy hold and updates the hold start date to the current date.

One week later, the hurricane has passed. The hurricane damaged Honolulu County only. The carrier updates the hold region to Honolulu County only. Because the carrier has not assessed the full extent of damages, the carrier is not yet ready to write policies in Honolulu County. Therefore, the carrier does not set the hold end date.

Three weeks later, the carrier has assessed all damage from the hurricane. The carrier sets the hold end date to the current date. The carrier leaves the policy hold in place to protect against someone trying to obtain insurance for the hurricane after the hurricane has passed. For more information, see “PolicyCenter Prevents Back-dating a Policy Transaction to Get Around an Underwriting Hold” on page 667.

If, on the other hand, the hurricane completely bypassed the state of Hawaii, the carrier disables or deletes the policy hold. If policy transactions had been held as a result of the policy hold, PolicyCenter automatically releases them and notifies the appropriate users.

## Policy Holds for Regulatory Changes

PolicyCenter provides policy holds based on the reference date for regulatory changes. The reference date type depends upon the policy hold definition. For more information, see “Hold Rules” on page 672.

Coverage forms and rates are constantly changing. Changes are generally made at the jurisdiction level by line of business. Although changes generally pertain to one jurisdiction, sometimes changes occur on a countrywide level. For example, several jurisdictions adopt the same change at the same time.

Changes in policy form patterns can occur due to updates made by ISO, a jurisdiction mandated coverage change, or a business decision to broaden or restrict coverage. Within the United States, rates changes most commonly occur on workers' compensation policies when NCCI sends out changes. Rate changes also occur when the business decides to adjust rates due to competition or poor loss results. When a rate or form change occurs on a specific date, the carrier cannot process all affected transactions until the change is approved and put into production. Generally work is put on hold prior to the date of the change so that the carrier does not start work with incorrect rates or forms.

For example, a company filed rates, but these rates are not yet approved. The filed rates affect upcoming renewals, so the company adds a policy hold that blocks the quote release until the rates are approved.

## Specifying Policy Holds

When adding a policy hold in PolicyCenter, you can specify the following:

- **The type of hold** – You can choose underwriting or regulatory.
- **The line of business** that the policy covers.

- **The policy transaction** – You can place policy holds on the following policy transaction types:
  - Issuance
  - Policy change
  - Reinstatement
  - Renewal
  - Rewrite
  - Rewrite new account
  - Submission
- **The written or reference date** that the policy hold starts and ends. You must enter the start date, but the end date is optional. To end a policy hold with no end date, simply edit the policy hold and specify an end date. For underwriting holds, you can specify either written date or effective date. For regulatory holds, you can only specify reference date.
- **The coverages** that trigger a policy hold.
- **The regions** that the policy hold affects. The policy hold can affect a policy if the policy has a location in the region. If you do not specify any regions, then PolicyCenter does not look at the policy locations.  
A policy may cover several locations. If one of the locations is in a region where a natural disaster is happening, then the policy hold may block policy transactions that are being performed on the policy.

In the default configuration, PolicyCenter checks to see if a policy hold applies at:

- Quote
- Bind
- Issue

If a policy hold applies, then PolicyCenter creates an underwriting issue which blocks progress of the policy transaction. For underwriting holds, the policy transaction blocks on bind or issue. PolicyCenter displays a warning on quote. For regulatory holds, the policy transaction blocks on quote.

When the policy hold no longer applies, the policy hold underwriting issue becomes an orphan on the policy transaction. PolicyCenter removes the orphaned underwriting issue from the policy transaction. If PolicyCenter removes a policy hold from an automated renewal, PolicyCenter returns the renewal to automated processing.

#### See also

- “Underwriting Authority” on page 411
- “Removing Orphaned Issues” on page 460 in the *Configuration Guide*

### [PolicyCenter Prevents Back-dating a Policy Transaction to Get Around an Underwriting Hold](#)

A carrier applies an underwriting hold to prevent new coverage being created in a region with an impending natural disaster. Potentially, an agent might try to circumvent an underwriting hold by back-dating the effective or written date of a policy transaction. For example, the agent starts a submission when a policy hold is in effect for submissions on that line of business. In an attempt to force PolicyCenter to issue the submission anyway, the agent sets the effective date of the submission to a date before the policy hold is in effect. If the agent issues the policy, the carrier has a high likelihood of having to cover claims on that policy.

To avoid this possibility, PolicyCenter determines that the underwriting hold is in effect if both of the following are true:

- The effective or written date of the object is less than the hold start date.
- The current date is greater than or equal to the hold start date.

**Note:** In the default configuration, PolicyCenter only checks for back-dating in underwriting holds. PolicyCenter does not check for back-dating in regulatory holds, which restrict the carrier from writing business pending regulatory approval.

# Working with Policy Holds

This topic provides step-by-step instructions for working with policy holds in PolicyCenter.

**Note:** You must have sufficient permission to view and work with the **Policy Holds** screen. For more information, see “Policy Hold Permissions” on page 675.

This topic has the following step-by-step instructions:

- “Example: Creating a Simple Policy Hold” on page 668
- “Working with Policy Hold Actions” on page 670
- “Deleting or Disabling a Policy Hold” on page 671

## Example: Creating a Simple Policy Hold

These step-by-step instructions show how to create a policy hold, view the policy hold on a policy transaction, approve the policy hold, and issue the policy.

You can follow these instructions that use the large sample data set. For information about loading the sample data, see “PC Sample Data” on page 139 in the *System Administration Guide*.

These instructions create an underwriting hold for renewals in the commercial property line of business. Note that in addition to underwriting issues for policy holds, other types of underwriting issues may block quote and bind.

This example has the following series of instructions:

1. “Create a Policy Hold” on page 668
2. “Create a Policy Effective in the Past” on page 669
3. “Renew Policies Through the Batch Process” on page 669
4. “Approve the Policy Hold” on page 670
5. “Issue the Policy” on page 670

### Create a Policy Hold

1. Log in as a user with permissions to view, create, and edit policy holds. If you are using the sample data, log in as su.
2. Select **Administration** → **Policy Holds**.
3. On the **Policy Holds** screen, click **Add**.
4. On the **Hold Details** tab, make the following selections:

Field	Value
Hold Type	Underwriting Hold
Code	UWHold_Hurricane001
Description	UW Hold Hurricane 001
Hold Start Date	Enter a date six months prior to the current date. The policy hold must start before the renewal effective date.
Hold End Date	Do not enter an end date at this time.
UW Issue Type	UW Policy Hold
Long Description	UW Issue for UW Hold Hurricane 001

5. In Hold Rules, click Add and make the following selections for the rule:

Field	Value
Line of Business	CP
Policy Transaction	Renewal
Policy Transaction Date Type	Effective Date

6. Click Update to create the policy hold.

### Create a Policy Effective in the Past

In the previous set of instructions, you created a policy hold for commercial property renewals. In these steps, you will create a commercial property policy for a one year term, starting two years ago. Because this policy started two years ago, it will be ready for renewal, and subject to the policy hold.

1. Log in as a user with permissions to create policies. If you are using the sample data, log in as the underwriter applegate.

**Note:** In general, a producer, such as aarmstrong, would create the new policy. However, aarmstrong does not work for this example because he does not have the ability to set the Written Date. Therefore, the example uses the underwriter applegate.

2. Go to an account. If you are using the sample data, go to the *Wright Construction* account.
3. Select Actions → New Submission.
4. Select Commercial Property.
5. On the Policy Info screen, set the Effective Date and Written Date of the policy to two years prior to the current date.
6. Continue the submission. If you are using the Wright Construction account, add a building on the Buildings and Locations screen:
  - a. In the Actions column for the location, click the arrow, and select Add Building → New Building.
  - b. On the New Building screen, enter a Property Class Code and Coverage Form.
  - c. On Coverages → Business Income Coverage, enter a numeric value in one of the Income Limit fields.
  - d. Click OK to create the building.
7. Click Quote.
8. Select Bind Options → Issue Policy.
9. Go to the Account Summary screen. Notice that the policy is expired because you created it effective two years in the past.

### Renew Policies Through the Batch Process

The Policy Renewal Start batch process creates renewal policy transactions for expired or soon to be expired policies. The batch process will create a renewal policy transaction for the policy you created in the previous set of instructions.

1. Log in as a user with permissions to access the Internal Tools → Batch Process Info screen. If you are using the sample data, log in as su.
2. Type ALT+SHIFT+T to view the Internal Tools → Batch Process Info screen.
3. In the Action column of the Policy Renewal Start batch process, click Run.

## Approve the Policy Hold

In the previous set of instructions, you ran the batch process that created a renewal policy transaction for the expired commercial property policy. In these instructions, you view the policy hold on the commercial property policy, and approve the underwriting issue for the policy hold.

1. Log in as a user with permissions to create policies. If you are using the sample data, log in as the underwriter **aapplegate**.
2. Go to the commercial property policy.  
On the **Summary** screen under **Pending Policy Transactions**, there is a Renewal policy transaction with **Quoted** status. The **Current Activities** shows a **Blocked - Pending Renewal** activity.
3. Click the activity link.
4. In the left sidebar of the renewal policy transaction, click **Risk Analysis**.  
The **UW Hold Hurricane 001** is blocking bind.
5. Select the check box for **UW Hold Hurricane 001** and click **Request Approval**.
6. Log out.
7. Log in as **ssmith**, who has the **Underwriter 2** authority profile on the **User → UW Authority** tab. An **Underwriter 2** can approve underwriting holds, but not regulatory holds.
8. Find the policy.
9. Click **Risk Analysis** in the left sidebar.
10. Select the check box for **UW Hold Hurricane 001**.
11. Click **Approve**.
12. Click **OK** on the **Risk Approval Details** screen.

## Issue the Policy

In the previous set of instructions, you approved the underwriting issue for the policy hold. In these instructions, you issue the policy.

1. Log in as **aapplegate**.
2. Go to the policy.  
Notice that the renewal policy transaction appears under **Pending Policy Transactions**.
3. Click the **Transaction #** link.
4. Select **Bind Options → Issue Now**.

## Working with Policy Hold Actions

These step-by-step instructions show you how the **Actions → Policy Holds** menu item affects a policy with a hold. For information about this menu item, see “Policy Hold Actions in Renewals” on page 673.

The instructions assume that you have loaded the large sample data set, and that you are creating a commercial property policy.

1. As **su**, create a policy hold for commercial property policy renewals as described in “Create a Policy Hold” on page 668.
2. As **aapplegate**, create a commercial property submission as described in “Create a Policy Effective in the Past” on page 669.
3. Create a second commercial property policy as described in step 2.

4. As su, run the **Policy Renewal Start** batch process as described in “Renew Policies Through the Batch Process” on page 669.

5. As aapplegate, go to each commercial property policy.

On the **Summary** screen under **Pending Policy Transactions**, notice that there is a Renewal policy transaction with Quoted status. The **Current Activities** shows a **Blocked - Pending Renewal** activity.

For each policy:

a. Click the **Blocked - Pending Renewal** activity link.

b. In the left sidebar of the renewal policy transaction, click **Risk Analysis**.

Notice that the **UW Hold 001** is blocking bind.

---

**IMPORTANT** The **Actions → Policy Holds** menu items apply to policy transactions in automated processing only. Therefore, you do not approve or reject the policy hold because this would remove the policy transaction from automated processing.

6. In the second commercial property policy, select **Actions → Policy Hold → Don't issue when hold released**.

7. Complete the activity.

8. As su, delete the policy hold:

a. Select **Administration → Policy Holds**.

b. Click the **Code** link for the policy hold.

c. Click **Delete**.

9. Type ALT+SHIFT+T. On the **Batch Process Info** screen, run the **Policy Hold Job Evaluation** batch process.

**Note:** In PolicyCenter, the user interface uses the term *policy transaction* to refer to submissions, policy changes, and other policy transactions. However, policy transactions are implemented as jobs in the data model, and referred to as jobs in PCF files, Gosu classes, and other configuration files. Therefore, the configuration documentation refers to policy transactions as jobs.

The batch process reevaluates both renewal policy transactions, and puts them back into automated renewal processing. The automated renewal processing binds and issues the first renewal policy transaction. Because this policy was issued one year ago, it expired today and appears in **Policy Terms** on the **Account File Summary** screen as an expired term.

The automated renewal processing quotes but does not bind or issue the second policy renewal policy transaction. The policy transaction appears under **Pending Policy Transactions** on the **Account File Summary** screen. The processing generates a **Policy Hold released** activity reminding the user to issue the renewal.

## Deleting or Disabling a Policy Hold

If you delete or disable a policy hold, PolicyCenter no longer applies the policy hold to policies.

- **To delete a policy hold** – Select the policy hold in the **Policy Holds** screen or view the policy hold, then **Delete** it.
- **To disable a policy hold** – Set the **Hold End Date** to the **Hold Start Date** to disable the policy hold.

When the policy hold no longer applies, the policy hold underwriting issue becomes an orphan on the policy transaction. PolicyCenter removes the orphaned underwriting issue from the policy transaction. If PolicyCenter removes a policy hold from an automated renewal, PolicyCenter returns the renewal to automated processing.

## Policy Hold Screen

You can access the screen for administering policy holds by navigating to **Administration** → **Policy Holds**. On this screen, you can add, copy, or delete a policy hold.

To create a policy hold, click **Add**.

To edit a policy hold, click the **Code** or **Description** of the policy hold. This action takes you to a details screen that has an **Edit** button.

**Note:** You must have sufficient permission to view, and work with the **Policy Holds** screen. For more information, see “Policy Hold Permissions” on page 675.

### Hold Details Tab

You can specify details of the policy hold on the **Hold Details** tab. This tab has the following fields:

Field	Description
Hold Type	Required. Select <b>Regulatory Hold</b> or <b>Underwriting Hold</b> . Your selection changes the availability of the <b>UW Issue Type</b> and <b>Long Description</b> fields. You cannot edit this field in a policy hold that already exists.
Code	Required. Enter a code for this hold. You cannot edit this field in a policy hold that already exists.
Description	Enter a description of the hold. This short description identifies the hold on the <b>Risk Analysis</b> screen.
Hold Start Date	Required. Enter a start date for the hold.
Hold End Date	Optional. Enter an end date for the hold. In some cases, you do not know the end date of a policy hold. For example, in an underwriting hold for a natural disaster, you do not know how long the disaster will last. In a regulatory hold for rate changes, you may not know when you will receive finalized rates. To end a policy hold that has no end date, simply edit the policy hold and specify an end date.
UW Issue Type	Required. The <b>Hold Type</b> selection determines the drop-down list choices. If the <b>Hold Type</b> is <b>Underwriting Hold</b> , then the only choice is <b>UW Policy Hold</b> . If the <b>Hold Type</b> is <b>Regulatory Hold</b> , then the only choice is <b>Regulatory Policy Hold</b> . The UW issue types are defined in the <b>UWIssueType</b> system table.
Long Description	Required. Provide a description for the policy hold. This field appears under the <b>Description</b> on the <b>Risk Approval Details</b> screen.

### Hold Rules

The hold rules section describes the rules for each policy hold. Each policy hold must have at least one hold rule. A hold rule specifies the line of business, policy transaction, date type, and coverage that PolicyCenter uses to determine whether or not to hold a policy transaction.

The following table describes the fields for hold rules:

Field	Description
Line of Business	Required. Enter the line of business to which this rule applies.
Policy Transaction Type	Required. Enter the policy transaction type to which this rule applies.
Transaction Date Type	Required. The <b>Hold Type</b> selection affects the drop-down list choices. If the <b>Hold Type</b> is <b>Underwriting Hold</b> , then the choices are <b>Effective Date</b> and <b>Written Date</b> . If the <b>Hold Type</b> is <b>Regulatory Hold</b> , then the only choice is <b>Reference Date</b> .
Coverage	Optional. Select a coverage from the chosen line of business.

If you add a rule for a regulatory hold, the **Transaction Date Type** is always **Reference Date**. The reference date type depends upon the rule definition. If you specify a coverage in the rule, then the reference date type is the reference date type of the coverage. If you do not specify a coverage, then the reference date type is the reference date type of the line of business.

#### See also

- “Policy Holds for Regulatory Changes” on page 666
- “Determining the Reference Date” on page 475

## Hold Regions Tab

You can optionally specify one or more regions that the policy hold covers. PolicyCenter compares the hold regions against the policy locations when processing a policy transaction.

When you click **Add Hold Region**, the **Search for Regions** screen appears. The choices on this screen depend upon the selected **Country**. For example, if the **Country** is **United States of America**, you can select a **Region Type** of **State**, **County**, **City**, and **ZIP code**. The **Region Type** determines the **Additional Search Criteria**.

The **Additional Search Criteria** allow you to narrow down a search of the specified **Region Type**.

## Copying a Policy Hold

The **Copy Policy Hold** drop-down list allows you to copy a selected policy hold. You can select to copy only the **Regions** of a policy hold, only the **Rules** of a policy hold, or both the **Regions** and **Rules** of a policy hold. The different choices on the drop-down list allow you to create a new policy hold that has some of the attributes from a prior hold.

## Policy Hold Actions in Renewals

If a policy hold blocks a renewal policy transaction, the **Actions → Policy Hold** menu appears for that policy transaction and has the following menu item:

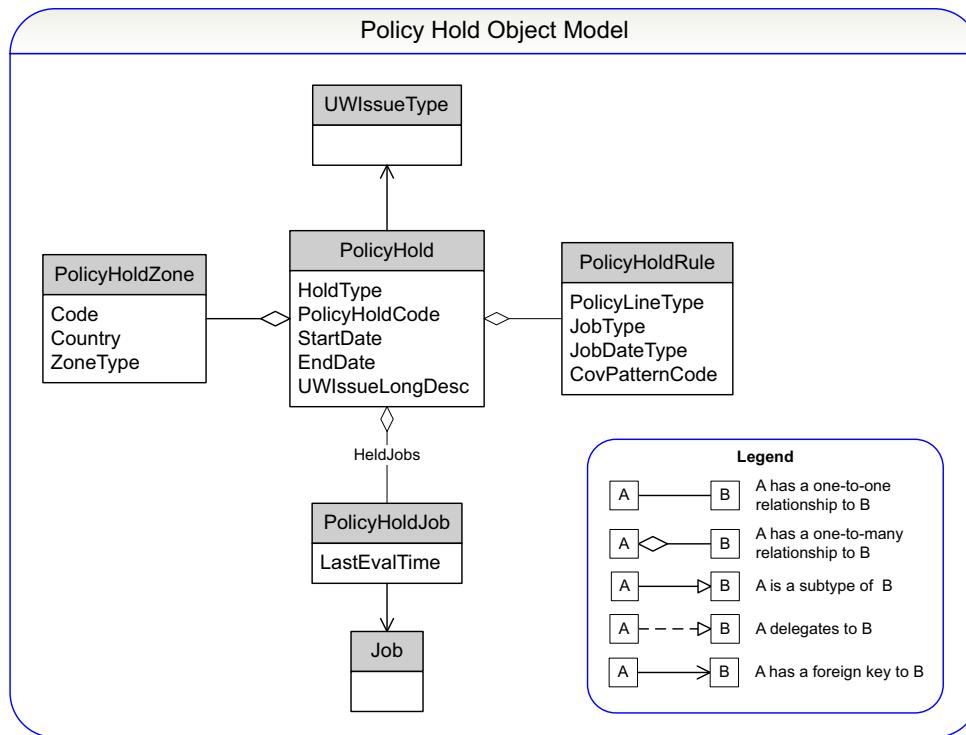
Menu item	Description
<b>Don't issue when hold released</b>	Do not issue the policy when PolicyCenter releases the underwriting hold.  When the <b>Policy Hold Job Evaluation</b> batch process runs, if the policy hold has been deleted or no longer applies, the renewal does not return to automated renewal processing. An activity is sent to remind the producer to retry issuing the renewal.
<b>Allow issuance when hold released</b>	This is the default setting. Issue the policy when the underwriting hold is released. When the <b>Policy Hold Job Evaluation</b> batch process runs, if the policy hold has been deleted or no longer applies, the renewal returns to automated renewal processing. PolicyCenter issues the policy when the renewal processing completes.

**Note:** If a renewal has been escalated to manual processing for another reason, then PolicyCenter does not return the renewal policy transaction to automated processing. This behavior applies even if the policy is set to **Allow issuance when hold released**. For example, assume the producer modified the renewal and adjusted a coverage limit without attempting to issue the renewal. PolicyCenter will not return the renewal policy transaction to automated processing even if the policy hold is released.

#### See also

- “Working with Policy Hold Actions” on page 670
- “Policy Hold Batch Process” on page 676

## Policy Hold Object Model



The **PolicyHold** entity is the main entity for policy holds. This entity has an array to access **PolicyHoldZone** entities which describe the regions for which the hold applies. This entity also has an array to access **PolicyHoldRule** entities which describe the rules for the hold. The rule entity specifies the line of business, job (policy transaction) type, date type, and coverages for which this rule applies.

The **PolicyHoldJob** entity keeps track of:

- Jobs that are currently under a policy hold.
- The last time PolicyCenter evaluated the job against the policy hold.

This entity has the following fields:

- **Job** – A foreign key to the job.
- **LastEvalTime** – The last time that PolicyCenter evaluated this job against this policy hold.

When a policy hold applies, PolicyCenter creates an underwriting issue of one of the policy hold types specified in the **UWIssueType** system table.

### See also

- “Policy Hold Underwriting Issue Types” on page 675
- “Underwriting Issue Object Model” on page 458 in the *Configuration Guide*

## Configuring Policy Holds

Policy holds are a type of underwriting issue. This topic describes how to configure policy holds.

### See also

- “Underwriting Authority” on page 411

- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*

## Policy Hold Permissions

The following permissions relate to policy holds.

Permission	Code	Description
Edit policy hold	polholdedit	Permission to edit a policy hold.
View policy hold	polholdview	Permission to access the <b>Policy Holds</b> screen.
Create policy hold	polholdcreate	Permission to create a policy hold.

In the sample data, the **Underwriting Supervisor** and **Superuser** roles have all the policy hold permissions. The **Underwriter** role has the **View policy holds** permission. For information about loading the sample data, see “PC Sample Data” on page 139 in the *System Administration Guide*.

## Policy Hold Authority Grant

Authority profiles determine the types of underwriting issues that a user can approve. Authority profiles contain authority grants. The **Regulatory Policy Hold** and **UW Policy Hold** authority grants to allow users to approve those types of policy holds.

### See also

- “Authority Profiles” on page 655

## Policy Hold Underwriting Issue Types

The **UWIssueType** system table defines issue types for underwriting and regulatory policy holds. You can view this table in Product Designer by navigating to **System Tables** → **uw\_issues\_types.xml**. The following table lists some of the values in the **UWIssueType** system table for policy hold underwriting issue types.

Code	Name	Blocking Point	Checking Set
UWPolicyHold	UW Policy Hold	Blocks Bind	Underwriting Hold
RegulatoryPolicyHold	Regulatory Policy Hold	Blocks Quote	Regulatory Hold

PolicyCenter evaluates each checking set at specific blocking points for each job (policy transaction) type. Then PolicyCenter determines whether or not to raise an underwriting issue.

There are two checking sets for policy holds: **Underwriting Hold** and **Regulatory Hold**. PolicyCenter evaluates these checking sets at all blocking points. Therefore, PolicyCenter notifies the user each time the job advances to the next step.

Blocking points in the **UWIssueType** system table represent points in the job at which an issue can block progress of the job. The underwriting hold blocks on bind, but regulatory hold blocks on quote. The **gw.job.JobProcessUWIssueEvaluator** class contains code that defines the blocking points at which to evaluate the checking set.

### See also

- “Underwriting Authority” on page 411
- “Configuring Underwriting Authority” on page 447 in the *Configuration Guide*

## Policy Hold Gosu Classes

In the `DefaultUnderwriterEvaluator` Gosu class, the `policyHold` method evaluates whether to raise underwriting issues. This class is in the `gw.lob.common` package.

This method checks to see if the checking set is either a regulatory hold or an underwriting hold. If so, the method retrieves all policy holds in the database, and compares each hold with the current policy period. If the details of the policy period match a hold, then the method raises an underwriting issue. The underwriting issue has the `Code`, `Description`, and `Long Description` of the policy hold.

The `policyHold` method also adds or updates the `PolicyHoldJob` element in `HeldJobs` array for this job (policy transaction) and policy hold. The method also updates the `LastEvalTime` field of that `PolicyHoldJob` to indicate the last time that PolicyCenter evaluated this hold against the job.

The policy hold batch process deletes the entry in the `HeldJobs` array that corresponds to this job and policy hold when the hold no longer applies to this job.

The code that compares the policy period with the hold is in the `gw.job.uw.PolicyHoldEnhancement` class. This enhancement contains methods that compares dates, locations, and coverages of the given policy period.

## Policy Hold Batch Process

The Policy Hold Job Evaluation batch process reevaluates all open policy transactions under a policy hold. If the policy hold conditions no longer apply, the batch process removes the policy hold from the policy transaction. PolicyCenter returns the renewal to automated processing except in the following cases:

- If a renewal has been escalated to manual processing, then PolicyCenter will not return the renewal policy transaction to automated processing. This behavior applies even if the policy is set to `Allow issuance when hold released`. For example, if a user approved or rejected a policy hold underwriting issue, then PolicyCenter will not return the renewal policy transaction to automated processing.
- If the policy transaction has not been escalated to manual processing but has `Actions → Don't issue when hold released`, the batch process sends the producer an activity reminder. The activity reminds the producer to retry quoting or binding the job.

In the default configuration, the Policy Hold Job Evaluation batch process runs nightly. You can also manually start the batch process from the **Batch Process Info** screen.

The code for this batch process is `PolicyHoldJobEval`. The batch process calls the `IPolicyHoldJobEvalPlugin`.

## Policy Hold Job Evaluation Plugin

The batch process calls the following methods in the policy hold job evaluation plugin (`IPolicyHoldJobEvalPlugin`):

- `findJobsToEvaluate` – Finds jobs that PolicyCenter needs to evaluate against the policy holds blocking the jobs. In the default configuration, this method finds jobs that are:
  1. Open
  2. Have a policy period with an active blocking policy hold.
  3. Has not been evaluated since the last time the policy hold has changed.
- `evaluate` – Determines if the policy hold was deleted or whether the policy hold still applies.

### See also

- “Batch Process Info” on page 120 in the *System Administration Guide*

## Importing and Exporting Policy Holds

Administrators can import and export policies under a policy hold.

**See also**

- “Importing and Exporting Administrative Data” on page 81 in the *System Administration Guide*



# Holidays and Business Weeks

Holidays, weekends, and business weeks define the PolicyCenter business calendar. The PolicyCenter business calendar calculates these dates and ensures the correct usage of holidays, weekends, and business weeks.

Some examples

- Activities usually reach their due dates and escalation dates after a defined number of business days. The activity patterns calculate the number of business days by using the holidays defined in the calendar.
- A regulatory agency specifies the maximum number of business days to perform an activity. The corresponding activity uses the holiday schedule to calculate the due date.

---

**IMPORTANT** Guidewire ClaimCenter enables you to specify holidays by zone, such as state and zip code, for use when assigning activities by location. Guidewire PolicyCenter does not provide support for zones.

---

This topic includes:

- “Specifying Holiday Dates” on page 679
- “Working with Holidays, Weekends, and Business Weeks” on page 680
- “Using Gosu Methods to Work with Holidays” on page 681
- “Business Weeks and Business Hours” on page 681
- “Holiday Permissions” on page 682

## Specifying Holiday Dates

In the base configuration, PolicyCenter determines weekends and work days by using configuration parameters in the `config.xml` file. However, you specify holidays through the user interface. Using the user interface gives you more flexibility in defining holidays, and you can make changes without having to restart the server.

To specify the holidays observed by your business, navigate in PolicyCenter to the **Administration tab → Business Settings → Holidays** screen. PolicyCenter stores all holidays you define in this screen in the database. All holidays are editable. With administrator privileges, you specify:

- **Name** – There is no limit on the holidays or on the names you give them. Each holiday is one day, so you must enter all the actual days if a holiday results in multiple days off. For example, you must specify two holidays for Thanksgiving in the United States if the company gives employees Thursday and Friday off.
- **Date** – The dates of some holidays vary each year, so this screen enables annual updates.
- **Applies to All Zones** – Determines who observes the holiday. You can further select the type of zone, such as state, county, or city in the United States if the holiday does not apply to all zones.  
PolicyCenter does not support this field.
- **Types** – Provides one way to categorize holidays. You can also define other types.  
The types in the base configuration are **Company Holidays** and **General**.

## Holiday Types

You can give holidays different classifications, or categories, by specifying the **Type** field. In the default configuration, PolicyCenter provides two types: **General** and **Company Holidays**.

You can use Gosu code in conjunction with holiday type to add logic to the handling of holidays. For example, if your company grants a holiday to all employees on the birthday of the company founder, you can create a Founders Birthday holiday of type **Company Holidays**. You can write code that avoids scheduling due dates on company holidays or add specific handling for special days like the Founders Birthday.

# Working with Holidays, Weekends, and Business Weeks

This topic describes how to work with holidays in the user interface.

---

**IMPORTANT** The **Applies to All Zones** field has no effect in PolicyCenter. The application always behaves as if **Applies to All Zones** is Yes.

---

### To add a holiday

1. Log in to the application as a user with administrator privileges, and then navigate to **Administration tab → Business Settings → Holidays**.
2. Click **Add Holiday** in the **Holidays** screen to create a new holiday. Enter the holiday name, date, and type into the screen.

### To edit a holiday

1. To edit a holiday, including its **Date**, **Type**, and **Zone**, click it on the **Holidays** screen, and click **Edit**.
2. Make your edits and click **Update** to save.

You can assign both **Type** and **Zone** to any choices that already exist, but you cannot create new choices for **Type** or **Zone** in this screen.

**Note:** You might need to change the **Date** of some holidays annually.

### To delete a holiday

You can delete any holiday on the **Holidays** screen. Select its check box and click **Delete**.

#### To create a new zone or type

In Guidewire Studio, navigate to the typelist that you want to modify.

- **Zone Type** – Defined by the `ZoneType` typelist, includes the typecodes `city`, `county`, `state`, `province`, `postalcode`, and `fsa`. You can add other types to this typelist.  
PolicyCenter does not support Zone Type. The application always behaves as if **Applies to All Zones** is Yes.
- **State** – Defines the states of the United States, Australia, and Germany, provinces of Canada, and prefectures of Japan that are in the `State` typelist.
- **Type** – Defined by the `HolidayTagCode` typelist. You can add other types to this typelist.  
The `HolidayTagCode` typelist includes the typecodes `General` and `CompanyHolidays`.

## Using Gosu Methods to Work with Holidays

You can write Gosu code to set business days differently for various tasks. For example:

- After auto-assigning a task to be completed in a certain number of business days, Gosu code can take into account the holiday schedule of the assignee.

Use Gosu methods that use `Holiday Type` and `Zone` to determine the correct number of business days.

#### Gosu Holiday Methods that Use Zones and Types

The methods `getConfiguredHolidays`, `addBusinessDays`, and `businessDaysBetween` on the `Date` entity get lists of holidays, add business days to dates, or compute business days between dates. Depending on the parameters, these methods can take into consideration holiday types or zones. You can find these methods in `gw.util.GWBaseDateEnhancement`, and you call them by using a `Date` object.

---

**IMPORTANT** Because PolicyCenter does not support zone, PolicyCenter ignores the `location` parameter. The application always behaves as if **Applies to All Zones** is Yes.

---

#### See also

- “Gosu Methods for Business Hours” on page 682

## Business Weeks and Business Hours

PolicyCenter can accommodate your business schedule by specifying your exact work week and hours. For example, the normal business hours of a carrier begin on Monday and end on Saturday. For this carrier, you configure PolicyCenter to have the hours from Monday to Friday begin at 8 a.m and end at 7 p.m. For Saturday, you configure the business hours to begin at 10 a.m and end at 2 p.m.

The `config.xml` file contains business calendar parameters. PolicyCenter applies these parameters system-wide. These parameters are the default values. For a complete list, see “Business Calendar Parameters” on page 38 in the *Configuration Guide*.

The business calendar parameters enable you to specify:

- For each day of the week, whether it is a business day. For example, to make Monday a business day, set `IsMondayBusinessDay` to `true`.
- The time that each business day starts and ends. Set `BusinessDayStart` and `BusinessDayEnd`.
- The day that is the end of the business week. Set `BusinessWeekEnd`.
- The time that marks the start of a new business day. Set `BusinessDayDemarcation`.

**See also**

- “Business Calendar Parameters” on page 38 in the *Configuration Guide*

## Business Hours

Business hours are defined in the `BusinessDayStart` and `BusinessDayEnd` configuration parameters. These times are based on the server clock. PolicyCenter provides Gosu methods that calculate elapsed hours by using these defined business hours. However, these defined hours do not deal with holidays accurately.

Specifying holidays affects only dates, not hours. However, you can write Gosu code for a task usually accomplished in hours rather than in days by using Gosu business hour methods. These methods take holidays into consideration after calculating business hours. They are completely separate from business day methods.

For example, a carrier promises to respond to all inquiries and claims within two hours after receiving an inquiry. You call the carrier on Friday at 4:30 p.m., and Monday is a holiday. The carrier must respond by Tuesday, one and a half hours after the business day starts.

### Gosu Methods for Business Hours

The methods `addBusinessHours` and `businessHoursBetween` on the `Date` entity add business hours to dates or compute business hours between specific dates. Depending on the parameters, these methods can take into consideration holiday types or zones. The methods also use the settings for business hours, days, and weeks in the `config.xml` file.

The methods are defined in `gw.util.GWBaseDateEnhancement`, and you call them by using a `Date` object.

### Notes

While certain methods might appear to be similar, they can have different results. For example,

- The method `addBusinessDays` works differently from `addBusinessHours`. For example, in the base configuration, a business day runs from 8:00 a.m. to 5:00 p.m. Adding one business day to Sunday 12:00 a.m. results in Monday 12:00 a.m. However, adding nine business hours to Sunday 12:00 a.m. results in Tuesday 8:00 a.m. In the base configuration, for calculation purposes, a business day includes the times 8:00 a.m. through 4:59 p.m. Therefore, adding 9 hours to a weekend day goes past the next business day, Monday, to 8:00 a.m. the following day, Tuesday.
- The method `businessDaysBetween` works differently from `businessHoursBetween`. If the business day is between 8:00 a.m. and 5:00 p.m., calling `businessDaysBetween` for Sunday 12:00 a.m. and Monday 12:00 a.m. returns a value of 1. Calling `businessHoursBetween` for Sunday 12:00 a.m. and Monday 12:00 a.m. returns 0.

## Holiday Permissions

The following system permissions control whether you can view the `Holidays` screen and edit the holidays:

- `holidayview`
- `holidaymanage`

In the base configuration, the Superuser role has these permissions.

# Policy Form Pattern Administration

This topic describes how to administer *policy form patterns*, which are physical documents that attach to a policy, in PolicyCenter. It is important to understand that PolicyCenter does not contain or store the content of the forms. Based upon the policy data, PolicyCenter simply identifies the forms associated with a policy.

This topic includes:

- “About Forms” on page 683
- “Working with Form Patterns” on page 685
- “Form Pattern or New Policy Form Screen” on page 686
- “Form Configuration” on page 693

**Note:** See “Forms Integration” on page 477 in the *Integration Guide* for more information on how to integrate forms with PolicyCenter.

## About Forms

PolicyCenter uses a *form* to represent a part of the policy contract. These aspects can include any and all of the following:

Form Type	Description
Declaration sheets	Forms that provide an index or summary of all exposures, coverages, and in some cases forms.
Policy definition forms	Forms that PolicyCenter associates with the policy, with a specific line of business, or coverages or coverables you select. Policy definition forms have language that defines – from a legal perspective – who is the insured, who is the insurer, and so on. Policy definition forms typically have a set of standard coverages that additional forms either amend or remove.
Coverage endorsements	Forms that add, remove, or clarify some type of coverage. For example, a Hired Auto Coverage Form might add hired auto coverage to a policy definition form that did not originally specify it.

Form Type	Description
Exclusion forms	Forms that limit coverages. For example, a Mold and Fungus Exclusion Form can limit coverage on a homeowner's policy. If included, it is possible that the policy does not cover any damage due to mold and fungus or perhaps cover it only to a certain amount.
Manuscript forms	Forms that are blank by default. The carrier can enter custom or special legal terms for the policy.

The insurance industry calls some types of forms *endorsements* if they extend the base policy contract form with additional language. Additionally, certain carriers call the policy change process in general *endorsing the policy*, because typical changes involve adding endorsements to the policy. Whether called forms or endorsements, forms are part of the legal contract between the carrier and the insured. In PolicyCenter, endorsements are simply one type of contract form.

As PolicyCenter issues a policy, it sends print requests for forms to the issuance system. These forms physically document the policy. Forms might exist in electronic form only, and the issuance system emails or faxes them to the insured. The issuance or document production system manages the actual content of a form.

**Note:** PolicyCenter does not store form content.

#### Differences Between Forms and Document Templates

PolicyCenter cannot add a form to a policy outside the context of a policy transaction. PolicyCenter can infer forms during any type of transaction. However, forms that PolicyCenter infers during audit policy transactions do not appear in the policy file after the transaction completes.

Outside the context of a transaction, you can use document templates for creating policy-related attachments. Both forms and document templates can initiate the creation of a document within the issuance system. The issuance system can upload the document to PolicyCenter, or PolicyCenter can contain a reference to that document. Forms and document templates have different tools and functionality associated with them.

- In PolicyCenter, the issuance system creates forms as part of the completion of a transaction.
- PolicyCenter can create document templates at any time in the policy life-cycle.
- Forms or document templates can be the basis of documents. Documents are the contents of either the form or document template. You can access documents directly from PolicyCenter after the issuance system creates them.

#### See also

- “Adding Document Templates to a Product” on page 21 in the *Product Model Guide*
- “Document Management” on page 453 in the *Application Guide*

## Form Basics

PolicyCenter does not store the content of the forms. PolicyCenter only identifies that one or more forms are associated with the policy. Within PolicyCenter, a forms screen lists form instances that indicate the physical forms that the issuance system creates. The form instance contains identifying information, such as the form number. The forms screen does not display the actual content of the form.

PolicyCenter automatically infers the necessary forms that the policy needs after you click **Quote** or **Bind**. In the base configuration, PolicyCenter identifies the forms to add:

- As it quotes the policy
- As it binds the policy

Therefore, in a policy transaction such as a submission, the **Forms** screen is unavailable before PolicyCenter quotes the policy. After PolicyCenter quotes the policy, you can access the **Forms** screen.

If you add additional options and re-quote the policy, PolicyCenter may determine that the policy needs more forms. During the binding process, PolicyCenter can determine that the policy needs additional forms as well.

#### See also

- “Policy Forms” on page 461 for information on forms in policy transactions.

## Working with Form Patterns

This topic describes how to work with form patterns in the user interface. You can view and create form patterns in the **Administration** tab of PolicyCenter.

### Searching for a Form Pattern

You can search for form patterns by form number or product.

**1. Select Administration → Business Settings → Policy Form Patterns.**

To view this menu item, you must have the **View form pattern** permission. The code for this permission is `formpatview`.

PolicyCenter displays the **Policy Form Patterns** screen.

**2. Enter a Form Number, Form Name, Product, or Group Code. Click Search.**

To search by **Form Number**, **Form Name**, or **Group Code** only, you must enter at least three characters in the text box. If you specify two or more fields, you can enter fewer than three characters in these fields.

**3. Click the form link in the Code column to view the Form Pattern screen for the form.**

**4. To duplicate or delete a form pattern, click the check box to the left of the Code column and click Duplicate or Delete.**

- Use the **Duplicate** drop-down list to create from one to 10 form patterns based on an existing one. Select the check box of the form pattern to copy, and select a number from the **Duplicate** drop-down list. Then edit only the settings that are different from the original. PolicyCenter enables the **Duplicate** drop-down if exactly one form pattern selected. You must have the **Create form pattern** permission to view **Duplicate** drop-down list. The code for this permission is `formpatcreate`.
- Use the **Delete** button to delete form patterns. You must have the **Delete form pattern** permission to view this button. The code for this permission is `formpatdelete`. Select the check box of one or more form patterns to delete, then click the **Delete** button. PolicyCenter enables the **Delete** button if you select at least one form pattern. You must have the **Delete form pattern** permission to view the **Delete** button. The code for this permission is `formpatdelete`.

**Note:** You cannot delete a form pattern when the server is in production mode. This restriction is to avoid deleting a **FormPattern** after generating a **Form** from it. PolicyCenter displays a validation error message if you try to delete a form pattern on a server in production mode.

### Adding a Form Pattern

**1. Select Administration → Business Settings → Policy Form Patterns.**

To view this menu item, you must have the **View form pattern** permission. The code for this permission is `formpatview`.

PolicyCenter displays the **Policy Form Patterns** screen.

**2. In the Search Results pane, click Add.**

You must have the **Create form pattern** permission to view these buttons. The code for this permission is `formpatcreate`.

PolicyCenter displays the **New Policy Form** screen.

3. Enter information about the form in the **New Policy Form** screen. See “Form Pattern or New Policy Form Screen” on page 686 for details about this screen.

## Specifying Removal or Replacement Forms for Policy Changes

You can specify a form to add when an associated form has been removed or replaced in a policy change.

### To specify the removal or replacement form pattern

1. Go to the **Form Pattern** screen for the removal or replacement form pattern.
2. Click **Edit**.
3. Go to the **Transaction Types** tab, and click **Add → Policy Change**.  
PolicyCenter displays a **Policy Change** tab.
4. Go to the **Policy Change** tab, and select **Yes** for **Is this form only used to indicate removal or replacement of another form?**

### To specify the associated form

1. Go to the **Form Pattern** screen for the associated form pattern.
2. Click **Edit**.
3. Go to the **Transaction Types** tab, and click **Add → Policy Change**.
4. Go to the **Policy Change** tab, and select **Yes** for **Can this form get added again on a policy change if its data changes....**
5. Go to the **If this form's data changes...** drop-down list. Select the removal or replacement form you defined in the previous step-by-step instructions.
6. Go to the **Inference** tab.
7. In **Form inference conditions**, select one of the following:
  - **An associated form is invalidated**
  - **An associated form is invalidated or replaced**

You must specify a **Start Date**, **End Date**, and **Edition** for the associated form.

### See also

- “Policy Change Tab for Form Patterns” on page 690

## Importing and Exporting Policy Form Patterns

You can import and export policy form patterns on the **Administration → Utilities → Import/Export Data** menu. You can use the import and export commands to transfer forms from a test environment to the production environment.

For instructions on how to use the **Import/Export Data** menu, see “Importing and Exporting Administrative Data from PolicyCenter” on page 87 in the *System Administration Guide*.

## Form Pattern or New Policy Form Screen

The **Form Pattern** screen displays details of the form pattern. The title of this screen is **New Policy Form** when you click **Add** on the **Policy Form Patterns** screen. PolicyCenter displays this screen when you add a new form pattern or when you click the link for an existing form pattern.

The **Form Pattern** screen has the following tabs:

- “Basics Tab for Form Patterns” on page 687
- “Products Tab for Form Patterns” on page 687
- “Transaction Types Tab for Form Patterns” on page 688
- “Jurisdictions Tab for Form Patterns” on page 688
- “Policy Change Tab for Form Patterns” on page 690
- “Inference Tab for Form Patterns” on page 692

## Basics Tab for Form Patterns

You can enter the following form pattern fields on the **Basics** tab:

Field	Description
Code	The form pattern code. This value must be globally unique. PolicyCenter stores this value in the <code>FormPatternCode</code> field of the <code>Form</code> and uses the value to link the form to a form pattern.
Number	The form number. Technically, this is an arbitrary string, but it typically is the form number used by the issuance system. This field is the form label that shows in the PolicyCenter user interface. PolicyCenter sets this value in the <code>FormNumber</code> field of a <code>Form</code> inferred from this pattern.
Edition	The form edition. PolicyCenter uses this value, in conjunction with the <code>Form Number</code> field, to indicate if two or more patterns are different editions of the same form. Similar to <code>Form Number</code> , this is technically an arbitrary string, but typically, it is the edition used by the issuance system. <b>IMPORTANT</b> If you add a new edition, you <i>must</i> also update the availability dates of the old and new patterns. Merely setting the edition is not sufficient.
Name	A human-readable description of the form pattern. By default, PolicyCenter sets this value in the <code>FormDescription</code> field of a <code>Form</code> during form inference.
Any data related to this form collected after quote?	If the answer is <b>No</b> , the <b>Quote</b> button in a policy transaction initiates forms inference. Quote-time inference happens immediately after the quote comes back and before it appears within PolicyCenter. Quote-time inference adds forms for which the form pattern specifies <i>either</i> quote-time or bind-time. This behavior is so that the user can see an estimate of the bind-time forms along with the quote. PolicyCenter removes and re-infers the bind-time forms when the policy transaction is bound. If the answer is <b>Yes</b> , the <b>Bind</b> button in a policy transaction initiates forms inference. Bind-time inference happens just prior to binding the branch. Bind-time inference removes any bind-time forms temporarily added during quote time. It then adds back only those forms for which the form pattern specifies bind-time.
Assign an endorsement number to this form?	Answer <b>Yes</b> if this form must have an endorsement number assigned to it at inference time. The endorsement number indicates the order in which the form is added to the policy. The carrier decides whether the form requires an endorsement number. Some carriers may require an endorsement number for forms added after issuing the policy. For example, a form added in a policy change requires an endorsement number.
Priority	An integer value that PolicyCenter uses to determine: <ul style="list-style-type: none"> <li>The order for assigning an endorsement number</li> <li>The processing order for forms during form inference</li> </ul> PolicyCenter processes and numbers the lower numbers first. PolicyCenter interprets a <code>null</code> value for this field as <code>MAX_INT</code> .
Integration Fields → Reference Code	Provided as a field for your customization. The base configuration of PolicyCenter does not use this field. You can use this field to specify a code for other systems in your enterprise to identify this form.

## Products Tab for Form Patterns

On the **Products** tab, you can select one or more products that use the current form pattern. For example, you can specify that the `Commercial Package` product uses `FormCPP01`.

On this tab, you specify all the products that can use this form pattern. The products can include both monoline products and package products. Sometimes package products use a form pattern, but only a particular policy line within the package actually uses the form pattern. In this case, use the **Inference** tab to specify the policy line that infers the form pattern.

## Transaction Types Tab for Form Patterns

On the **Transaction Types** tab, you can select the policy transaction types that use this form pattern. The types that you can select are:

- All Issuance and Change Transaction Types
- Audit
- Cancellation
- Policy Change
- Reinstatement
- Renewal
- Rewrite
- Rewrite New Account
- Submission

When you choose **All Issuance and Change Transaction Types**, PolicyCenter adds the **Submission**, **Renewal**, **Rewrite**, **Policy Change**, and **Rewrite New Account** policy transaction types to the form pattern. If these five policy transaction types are already on the form pattern, then **All Issuance and Change Transaction Types** is not visible. If some of the five policy transaction types are on the form pattern and you select **All Issuance and Change Transaction Types**, then PolicyCenter adds the remaining policy transaction types.

Although you cannot choose the issuance policy transaction, PolicyCenter uses forms specified for the submission policy transaction for both submission and issuance policy transactions.

When you select a policy transaction (job), PolicyCenter adds it to the **FormPatternJobs** array on the **FormPattern** entity.

## Jurisdictions Tab for Form Patterns

The **Jurisdictions** tab has two panels:

- Availability Table
- Jurisdictional Replacements

### Availability Table

In the **Availability Table** panel, you can **Add**, **Duplicate**, and **Remove** rows. Each row has the following fields:

Field	Description
Available	Required. Select Yes or No.
Start Date	The start date is inclusive.
End Date	The end date is exclusive.
Jurisdiction	Select a jurisdiction. You can select only one jurisdiction per row.
UW Company	Select an underwriting company. You can select only one underwriting company per row.

PolicyCenter displays the availability rows in evaluation order. PolicyCenter determines if a form pattern is available for a policy as follows:

1. Search the rows from top to bottom until a match is found for the given jurisdiction and underwriting company. PolicyCenter groups rows with the same jurisdiction and underwriting company, but different date ranges. If no matches are found, the form is not available.
- Note:** A null value matches any input for that column.
2. Within the group of matching rows, PolicyCenter finds the row with a date range containing the given reference date.
  - If a match is found, the **Availability** column determines whether the form is available.
  - If no match is found, the form is not available.

## Jurisdictional Replacement

Jurisdictional replacement is the concept of one form replacing another form in a particular jurisdiction. For example, certain jurisdictions have a jurisdiction-specific versions of a form. For example, California and Kansas have jurisdiction-specific version of the same form. There is also a general U.S. version of the form that all other jurisdictions use.

The **Jurisdictional Replacements** pane has the following fields:

Field	Description
<b>Group Code</b>	Groups together a set of forms that, for jurisdictional replacement reasons, need to be processed together. PolicyCenter processes all forms that are part of the same group code together. If no <b>Group Code</b> is specified, PolicyCenter uses the <b>Form Number</b> as the <b>Group Code</b> of the form.
<b>This form is the jurisdiction-specific version of:</b>	In the jurisdictions available for the current form pattern, specify the other form pattern that the current form pattern replaces. This drop-down list displays all forms with the same group code as the current form.

The form inference in PolicyCenter processes all form patterns with the same group code together. At inference time, PolicyCenter determines the set of jurisdictions for which each pattern within the group is available.

PolicyCenter uses the **This form is the jurisdiction-specific version of** links to determine the appropriate form pattern in the group for each covered jurisdiction. PolicyCenter associates each covered jurisdiction with the most appropriate form pattern in the group. For a U.S. policy, this process lets PolicyCenter determine which of the covered jurisdictions use:

- The U.S. version of the form
- A jurisdiction-specific version of the form

PolicyCenter then determines the appropriate data to populate the form, such as to populate a list of buildings in the jurisdictions for which the form applies.

The form inference in PolicyCenter may find no jurisdictions available for a form pattern within the group. For example, the form pattern is a U.S. form, and there are jurisdiction-specific forms for all the covered jurisdictions on the policy. If this is the case, PolicyCenter does not process the U.S. form any further, because the jurisdiction-specific forms replace the U.S. form.

The form inference may find one or more jurisdictions available for a form pattern within the group.

PolicyCenter uses the form for the set of jurisdictions in which the form is necessary. PolicyCenter removes from the set any jurisdictions that replacing forms cover.

The form replacement algorithm is recursive. For example, if form 16\_1\_CT is used instead of form 16\_0\_CT\_TX, which in turn is used instead of form 16\_0\_US, the end result is the following:

- Form 16\_1\_CT is available in CT.

- Form 16\_0\_CT\_TX is available in TX.
- Form 16\_0\_US is available everywhere but CT and TX.

## Policy Change Tab for Form Patterns

The **Policy Change** tab contains settings for the form in policy change transactions. PolicyCenter displays this tab if you added **Policy Change** to the **Transaction Types** tab.

The **Policy Change** tab has the following fields:

Field	Description
Is this form only used to indicate removal or replacement of another form?	Values are Yes and No. This input is always visible. This field indicates whether the form is a removal endorsement. The RemovalEndorsement field on the FormPattern entity stores the value of this field.
Can this form get added again on a policy change if its data changes (i.e. replaced)?	Values are Yes and No. This field is visible if the value of the first field is No. This field indicates whether to reissue the form when its inference data changes. The ReissueOnChange field on the FormPattern entity stores the value of this field.
If this form's data changes, also add the following invalidation/replacement form:	Value is a drop-down list of form patterns to add or remove because the data has changed. PolicyCenter populates the drop-down list with all removal endorsements valid for the same policy line. This field is visible if the value of the second field is Yes.

## Reissuing Forms During Policy Changes

Set the answer to **Can this form get added again on a policy change if its data changes...?** to Yes to reissue forms on a policy.

The form can be initially added to a policy during a submission, issuance, rewrite, renewal, or policy change transaction. If there is a change to the underlying data, the form can be reissued in a policy change transaction.

During a policy change, PolicyCenter must determine whether a version of the reissued form is already present. If it is, it must determine whether the data on the old version of the form is the same as the new version of the form. PolicyCenter compares the data between the old and new versions of the form. PolicyCenter compares:

New version of form	Old version of form
The XML created by the addDataForComparisonOrExport method of the corresponding forms inference class from the new version of the form	To the equivalent XML persisted with the old version of the form.

There are multiple possible outcomes:

### If a previous version of the form exists...

- ... And the form pattern is no longer inferred by the current data, PolicyCenter does the following:
  - Sets the removal date on the previous version so that PolicyCenter ignores it during subsequent forms processing.
  - If the pattern specifies adding another form when the current form is inaccurate because the data has changed, PolicyCenter infers a copy of that other form pattern.

PolicyCenter also performs these actions if there is no new version of the form at all.

- ... And PolicyCenter infers the form pattern by the current data and the form contains exactly the same data as the previous version, PolicyCenter does nothing with the form.
- ... And PolicyCenter infers the form pattern by the current data and it does not contain exactly the same data as the previous version, PolicyCenter does the following:
  - Sets the removal date on the previous version.
  - It adds the new version.

- It marks the new version as superseding the old version.
- If the pattern specifies adding another form when the current form is inaccurate because the data has changed, then PolicyCenter infers a copy of that other form pattern.

**If no previous version of the form exists...**

- ... And PolicyCenter infers the form pattern by the current data, PolicyCenter adds the new version of the form.
- ... And PolicyCenter does not infer the form pattern by the current data, PolicyCenter does nothing with the form.

### Do not Reissue

Set the answer to **Can this form get added again on a policy change if its data changes...?** to **No** to specify a one-time form on a policy.

These are forms that PolicyCenter adds to the policy at most a single time during a policy term and that never require updating or removal. A one-time form is generally suitable for use with a form that does not have any policy-specific data (for example, for something like a jurisdiction notice). PolicyCenter adds a one-time form if the `InferredByCurrentData()` property on `FormData` returns `true` and if there is no previous version of the form with the current policy term.

### Removal Endorsement

*Removal endorsements* are forms that nullify a previously issued form. Set the answer to **Is this form only used to indicate removal or replacement of another form?** to **Yes** to specify a removal endorsement form on a policy.

Removal endorsements have the unique property of always being processed last. Therefore, these endorsements can refer back to any forms that PolicyCenter removed earlier in the inference process. You use removal endorsements to indicate that a particular contract form is no longer valid.

You specify a removal endorsement by specifying to **Yes** to the question **Is this form only used to indicate removal or replacement of another form?**. In addition, a removal endorsement is typically configured by choosing one of the following answers to the **Form inference conditions** field on the **Inference** tab:

- An associated form is invalidated
- An associated form is invalidated or replaced

## Inference Tab for Form Patterns

The Inference tab has the following fields:

Field	Description
Policy line	Select a policy line. Because package policies are products, they do not appear in the drop-down list.
Form inference conditions	<p>Values are:</p> <ul style="list-style-type: none"> <li>• &lt;none selected&gt;</li> <li>• <b>An associated form is invalidated</b> – In a policy change, add this form if an associated form is no longer valid. A form is invalid if this form nullifies a form previously added to the policy. For example, a form was added to a policy because the policy had comprehensive coverage. In a policy change, this form is added if the comprehensive coverage is removed.</li> <li>• <b>An associated form is invalidated or replaced</b> – In a policy change, add this form if an associated form: <ul style="list-style-type: none"> <li>• Is no longer valid.</li> <li>• Was replaced by an updated copy of the same form. For example, if the schedule of items printed on the form was updated.</li> </ul> </li> <li>• <b>No additional criteria</b> – Always add this form as long as the product, policy transaction, and jurisdiction match.</li> <li>• <b>Selected coverage, condition, or exclusion is used</b> – Add this form if the selected coverage, condition or exclusion exists on the policy. You can choose whether the selected clause must exist on all instances of the associated coverable or just any instance.</li> </ul> <p>In addition, you can specify that a policy change will update the form if there are changes to any of the selected:</p> <ul style="list-style-type: none"> <li>• Coverage terms</li> <li>• Fields in covered objects</li> </ul> <p>• <b>Selected covterm value is used</b> – Similar to the previous field, but you can specify one or more values for option, package, or typelist coverage terms. PolicyCenter infers this form if one of the specified coverage term values has been selected on the policy.</p> <p>In addition, you can specify that the form will be updated in a policy change if there are changes to any of the selected:</p> <ul style="list-style-type: none"> <li>• Fields in covered objects</li> </ul> <p>• <b>Selected typelist value is chosen</b> – Add this form if the value set for a typekey field of a coverable in the policy data matches the selected value. You can choose:</p> <ul style="list-style-type: none"> <li>• Any coverable on the selected policy line.</li> <li>• Any typelist field on that coverable.</li> <li>• Any value on that typelist. You can choose only one value.</li> </ul> <p>You can choose whether the selected value must be set on all instances of the coverable or just any instance.</p>

The **Form inference conditions** field displays a drop-down menu which contains the `DisplayName` of all classes that implement the `gw.forms.GenericCenterFormInference` interface.

If you have a form that requires inference logic beyond what you can specify in the **Inference** tab, you can specify a custom form inference class. The custom form inference class overrides any previous settings on the **Inference** tab. If a form has a custom inference class, the **Infer this form when these match** field does not appear. The screen displays a message that this form uses custom inference logic defined in Guidewire Studio.

### See also

- “Configuring Generic Form Inference” on page 693
- “Reissuing Forms During Policy Changes” on page 690

## Form Configuration

In the Guidewire product model, a `FormPattern` describes the conditions for the attachment of any particular form to a product or a policy line. You define form patterns in the **Administration → Product Definition → Policy Form Patterns** tab of PolicyCenter. PolicyCenter associates a form pattern with a variety of criteria including product, policy line, policy transaction, and jurisdiction. The Inference engine uses the form pattern to determine whether to associate this particular form with the policy.

### Form Pattern Validation

When you click the **Update** button to save changes to a form, the page calls the static `FormPatternValidation.validateFormPattern` method, which performs several consistency checks. If there are no problems with the form, PolicyCenter saves your changes. Otherwise, PolicyCenter displays warning messages and does not save your changes.

You can validate all form patterns by selecting **Actions → Check policy form patterns for errors**. This action can be useful when making changes to one form pattern that may affect other form patterns. An example of a change to one form pattern is deleting an existing removal endorsement from a test environment.

The `FormPatternValidation` class has the `@Export` annotation, so you can modify the validations as necessary.

### Configuring Generic Form Inference

Forms inference determines the forms that PolicyCenter adds to a branch.

You can configure the inference conditions for most forms in the **New Policy Form** screen without writing any custom Gosu code. You configure generic inference on the **New Policy Form** screen. The inputs to the generic inference process are:

- The data entered on the **Form Pattern** or **New Policy Form** screen, including the data on the **Inference** tab
- The policy data in the branch

The output of the inference process is a list of form objects to add to the branch. PolicyCenter adds forms through all bound branches from bound policy transactions for the policy term. Cumulatively, these forms describe the policy contract and additional policy communications. The forms inference process evaluates all form patterns defined for the product and policy lines to determine which forms are implied by the policy data on the current branch.

You can add your own generic inference types. For example, you can add a type that infers forms based upon the presence of a particular coverable entity.

**Note:** PolicyCenter performs forms inference by using the Inference engine defined in `gw.forms.FormInferenceEngine`.

### Configuring Custom Form Inference

A *custom form inference class* is a subclass of `gw.forms.FormData` that does not implement the `GenericFormInference` interface. You cannot associate a form with a custom form inference class in the **Policy Form Patterns** screen. Instead, you must modify the `custom_form_inference` system table in Studio. Each row in the table corresponds to a particular form pattern code. Multiple rows in the table can reference the same custom inference class for different form pattern codes.

You can **Add**, **Duplicate**, or **Remove** rows in the `custom_form_inference` table.

For each row in the table, you must enter:

Field	Description
FormCode	The code for the FormPattern. Enter the same value into the Code field on the Basics tab of the policy form screen.
InferenceClass	Enter the fully-qualified name of the inference class. To help you enter the inference class, you can click the SmartHelp light-bulb icon to see a drop-down menu of all available inference classes.

For all form patterns listed in the `custom_form_inference` table, PolicyCenter uses the custom inference class. If this table does not list a form pattern, PolicyCenter uses the generic inference class specified in the `GenericInferenceClass` field of the `FormPattern`.

### Custom Inference Class

A custom form inference class is a subclass of the `FormData` abstract class that links to each form. For example, abstract class `WCFormData` extends `FormData` and provides some helper methods which Workers' Compensation uses. In turn, the `Form_WC_00_06_03_CT` class extends `WCFormData`. In the `custom_form_inference` table in Studio, the definition of form `WC_00_06_03_CT` sets `gw.lob.wc.forms.Form_WC_00_06_03_CT` as the class to use in processing the form logic for that particular form.

Multiple form patterns can share an inference class if appropriate.

The `FormData` class is an abstract class that provides common functionality such as populating the `FormDescription` field on `Form`. By default, PolicyCenter uses the description of the pattern. It also contains methods for populating additional fields on the form. If you extend `FormData` class, then you must implement the following abstract methods in your subclass:

- `populateInferenceData`
- `InferredByCurrentData`
- `addDataForComparisonOrExport`

The `FormData` class contains a default implementation of `getLookupDate` which uses the reference date of the current policy transaction as the lookup date for determining which form patterns are available. In the case of forms directly related to a coverage, condition, or exclusion, you can override the `getLookupDate` method. The override uses the reference date of that coverage, condition, or exclusion to determine form pattern availability. The reference date of the coverage, condition, or exclusion is potentially different than the reference date of the current policy transaction.

If PolicyCenter finds that a form is available in at least one jurisdiction by using the lookup date returned by `getLookupDate`, it then calls its `populateInferenceData` method. Input parameters to this method are the inference context object and the set of available jurisdictions. Typically, the `populateInferenceData` method looks through the policy data and collects the information relevant to that form. If `getLookupDate` is overridden, then `getLookupDate` will likely perform that work and `populateInferenceData` may do nothing.

After PolicyCenter populates the data, the inference engine invokes the `InferredByCurrentData` property on the form. This property determines whether the form is part of the policy based on the current state of the data. However, even if the method returns `true`, it does not mean that PolicyCenter expects the issuance system to issue the form. The field values on the form pattern and the data for comparison or export must still be considered.

For example, consider a form pattern that you do not want added again mid-term if its data changes. PolicyCenter issues the form only if the form does not already exist on the policy within the current policy term. Or, consider a form pattern that you want added again when its data changes mid-term. PolicyCenter issues the form only if the form would contain changed data compared to the last time it was added to the policy. As such, returning `true` from `InferredByCurrentData` indicates that the policy data on the policy transaction satisfies the

conditions for including the form on the policy. This method does not take into account the policy lifecycle or previously issued forms.

**Note:** See “Forms Integration” on page 477 in the *Integration Guide* for more information on forms inference.

## Adding a Custom Inference Class

Follow these steps to add a form pattern that uses a custom inference class. In this example, you define a form for the jurisdiction, Kansas, that has a custom inference class. The form code is WC\_00\_06\_03\_KS. The inference class is gw.lob.wc.forms.Form\_WC\_00\_06\_03\_KS.

### Define the inference class

1. Determine the name of the custom inference class. You can use an existing custom inference class or create a new one. In this example, the form uses the gw.lob.wc.forms.Form\_WC\_00\_06\_03\_KS inference class.
2. In Studio, navigate to the custom inference class. If the class does not exist, then create it. For this example, navigate to **configuration → gsrc**. In the gw.lob.wc.forms package, right-click to select **New → Gosu Class**, enter **Form\_WC\_00\_06\_03\_KS**.
3. Add code to handle inference for your form. See “Custom Inference Class” on page 694 for details on the methods you must create.

### Add the form to the custom form inference table in Product Designer

1. In Product Designer, open **System Tables → custom\_form\_inference.xml**.
2. Click **Add** and enter the form code in the dialog. For this example, enter **WC\_00\_06\_03\_KS**.  
Product Designer adds the form code to the table.
3. In **Inference Class**, add the fully qualified path to the custom inference class. For this example, enter **gw.lob.wc.forms.Form\_WC\_00\_06\_03\_KS**.

### Define the form pattern in PolicyCenter

If you are connected to the PolicyCenter server, you can synchronize the system table changes from your change list. These steps assume that you have loaded the large sample data set. For more information, see “Installing Sample Data” on page 52 in the *Installation Guide*.

1. In Product Designer, select **File → Synchronize System Tables**.
2. In PolicyCenter, log in as a user with permissions to add forms. If you have installed the sample data, you can log in as **svisor** with password **gw**.
3. Select **Administration → Business Settings → Policy Form Patterns**.  
PolicyCenter displays the **Policy Form Patterns** screen.
4. In **Form Number**, enter **WC 00**. In **Product**, select **Workers' Compensation**. Click **Search**.  
Notice that there are two forms with **Group Code** of **WC 00 06 03**. In the following steps, you will add a form with this group code.
5. Under **Search Results**, check **WC\_00\_06\_03\_CT**. Click **Duplicate**.  
PolicyCenter displays the **New Policy Form** screen.

6. On the Basics tab, enter the required fields, including these values:

Field	Value
Code	WC_00_06_03_KS
Number	WC 00 06 03 KS
Edition	Enter a month and year, such as 12 10.

7. On the Products tab, select Add → Workers' Compensation.  
 8. On the TransactionTypes tab, select Add and add Policy Transaction Types.  
 9. On The Policy Change tab, enter the following values:

Field	Value
Is this form only used to indicate removal or replacement of another form?	No
Can this form get added again on a policy change if its data changes...?	Yes
If this form's data changes, also add the following invalidation/replacement form:	WC 89 06 14

10. On the Inference tab, for Policy line, select Workers' Comp Line. This setting must match the other form patterns for this group code.  
 Also verify that the Inference tab displays the message that this form uses custom inference logic defined in Guidewire Studio.

11. On the Jurisdictions tab, enter the following values:

Field	Value
Jurisdiction	Select Kansas
Group Code	WC 00 06 03
This form is the jurisdiction-specific version of:	Select WC 00 06 03 - 08 05 [WC_00_06_03_US]

**Note:** You must have selected Workers' Comp Line in step 10 to see this value in the drop-down list.

12. Add any addition information about the form. Click Update.



## chapter 56

# Administering Policy Data Spreadsheet Import/Export

In PolicyCenter, policy data spreadsheet import/export enables you to export policy data to and from a spreadsheet. You can review and revise the exported data in a spreadsheet editor. You can import data from a spreadsheet into PolicyCenter.

This topic provides information about spreadsheet export formats, which define the columns that PolicyCenter exports when the user exports a spreadsheet template.

This topic includes:

- “Spreadsheet Export Formats User Interface” on page 697
- “Defining Export Formats” on page 697

**See also**

- “Policy Data Spreadsheet Import/Export” on page 463
- “Configuring Policy Data Spreadsheet Import/Export” on page 483 in the *Configuration Guide*

## Spreadsheet Export Formats User Interface

You can define export masks for policy data spreadsheet import/export on the **Spreadsheet Export Formats** screen of the **Administration** tab. To access this screen, you must have the **Manage export masks** permission. The code for this permission is `exportmasksmanage`. This permission enables administrators to restrict access to this screen. In the default configuration, only to the **Superuser** role has this permission.

## Defining Export Formats

Export formats define a subset of columns to export. Administrators can separately define export formats for commercial property locations and commercial property buildings. You can designate one format of each type as the default.

**Note:** A format can only remove columns from the spreadsheet. Adding columns is a configuration activity described in “Adding Spreadsheet Import/Export to Other Entities” on page 484 in the *Configuration Guide*.

**To define an export format**

1. Click the **Administration** tab and select **Spreadsheet Export Formats** to display the **Spreadsheet Export Formats** screen.
2. Click **New** to display the **New Export Format** screen.
3. Make the appropriate selections as described in the following table.

Field	Description
Name	Type a name for the format.
To Export	Select either <b>Commercial Property Locations</b> or <b>Commercial Property Buildings</b> . Each format applies to one of these coverable types.
Columns Included by Default	Lists the columns that are always included in the exported spreadsheet. You cannot omit these columns.
Available Columns	The list on the left contains the columns that are available to include in the exported spreadsheet. The list on the right contains the columns that are currently included in the format definition. To exclude columns, select them in the list on the right and click <b>Remove</b> . To include columns, select them in the list on the left and click <b>Add</b> .

4. Click **Update** to save the format and return to the **Spreadsheet Export Formats** screen.
5. To set a format as the default format, select its check box and click **Set Default**. When you export a spreadsheet, the default format is initially selected in the **Format** list.

# Product Integrations



# Billing System Integration

The default configuration of PolicyCenter includes a completely functional integration with Guidewire BillingCenter. You can also integrate PolicyCenter with the billing system of your choice. This topic describes how PolicyCenter integrates with a billing system in general, and BillingCenter in particular.

In the default configuration, PolicyCenter is set up to exchange information about accounts, policies, producers and producer codes, and billing with a billing system. The billing system is BillingCenter if you enable the integration.

The payment screen in PolicyCenter displays billing and payment methods retrieved from the billing system. PolicyCenter transmits the billing and payment method that you select to the billing system and saves it with the policy period. PolicyCenter and the billing system share accounts and policy periods. PolicyCenter receives delinquency notices from the billing system. If you enable the BillingCenter integration, the PolicyCenter user interface displays links that enable you to view data in BillingCenter.

This topic includes:

- “Billing System Integration Overview” on page 702
- “Policy Transactions That Create a New Policy Period” on page 705
- “Policy Transactions That Create Midterm Changes” on page 707
- “Working with the Billing System Integration” on page 711

## See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide* for detailed instructions on how to enable the integration.
- “Billing Integration” on page 509 in the *Integration Guide* for detailed information on how PolicyCenter integrates with BillingCenter or other billing systems.

## Billing System Integration Overview

The integration between PolicyCenter and the billing system allows the applications to share information by sending information through a plugin. The BillingCenter integration plugin sends and receives web service messages from BillingCenter.

Both PolicyCenter and the billing system maintain shared account, policy period, billing, and other information. While both applications have access to the information, only one application is the *system of record* (SOR) for each piece of information. In the BillingCenter integration, either PolicyCenter or BillingCenter is the system of record. The integration handles shared information in the following ways:

- You create and edit most shared information in PolicyCenter then push the information to BillingCenter.
- In some cases, an initial value is set in PolicyCenter so that BillingCenter automatically creates a new account or policy. After that, BillingCenter owns the value.
- You can customize the integration to use another application as the system of record. This application provides shared information to both PolicyCenter and the billing system. For example, a carrier can have a producer management system which manages producers and producer codes for PolicyCenter and the billing system.

PolicyCenter enables you to view billing information retrieved from the billing system. PolicyCenter displays billing information for the convenience of users who work mostly in PolicyCenter or do not have access to the billing system. If you have a BillingCenter login, PolicyCenter provides links to view account and policy period information in BillingCenter.

This topic contains the following topics:

- “Organizations and Producer Codes Overview” on page 702
- “Account Overview” on page 703
- “Policy Overview” on page 704
- “Sending Charges and Other Information to Billing” on page 704
- “Contact Overview” on page 704

### Organizations and Producer Codes Overview

In PolicyCenter, an organization contains producer information. The billing system must have an equivalent to organization. The BillingCenter equivalent to an organization is a producer. Both PolicyCenter and the billing system must have producer codes. What is the importance of a producer code?

- Every policy has one.
- It controls security in PolicyCenter.
- It identifies the producer for commission and agency bill.
- It identifies the commission plan.

The billing system pays the commission to the producer who holds a particular producer code. Therefore, every producer code has an owning producer. The producer code also provides the link to the commission plan. The producer also receives notices related to agency billing.

In PolicyCenter, users enter and manage organizations and producer code information. PolicyCenter propagates organizations and producer codes to the billing system when you create new or update existing organizations and producer codes. The updates contain the fields that are of interest to the billing system.

PolicyCenter propagates the internal carrier organization on start up. When PolicyCenter starts, PolicyCenter checks to see if the organization is a producer. If so, PolicyCenter sends a message asking BillingCenter to check whether it already has a producer for the internal carrier organization. If not, then PolicyCenter sends a message asking BillingCenter to create the producer for the internal carrier organization.

If you want to share organizations and producer codes between PolicyCenter and the billing system, you must create them in PolicyCenter.

#### See also

- “Billing Producers and Producer Codes” on page 515 in the *Integration Guide*

## Account Overview

Both PolicyCenter and the billing system must have accounts. For accounts that both systems share, there must be a one-to-one mapping between accounts in both applications. However, the structure and fields in the account entity may differ between the two applications. Both PolicyCenter and BillingCenter have account entities.

When you complete the first submission for a new account, PolicyCenter sends a message to the billing system to creates an equivalent account. When an account changes, PolicyCenter propagates those changes to the billing system. In the BillingCenter integration, the new BillingCenter account has default values for certain fields such as billing and delinquency plans which require an initial value.

If you want to share an account between PolicyCenter and the billing system, you must create the account in PolicyCenter.

Subaccounts and invoice streams provide flexibility for billing policies on the account.

## Subaccounts for Billing

An account can have multiple subaccounts for billing policies on the account. For example, a company has one subaccount for paying workers’ compensation policies, and another subaccount for paying business auto premiums. An accountant at the company can easily see the billing information for each type of policy.

In the BillingCenter integration, subaccounts are accounts that mark the current account as their parent in BillingCenter.

In the default integration, PolicyCenter does not have a screen for specifying a billing subaccount. The integration retrieves billing subaccounts from a billing system. In the default integration, the `StandAloneBillingSystemPlugin` simulates retrieving billing subaccounts from a billing system. The large sample data set has examples of billing subaccounts. For information on how to load the sample data, see “Installing Sample Data” on page 52 in the *Installation Guide*.

**Note:** Billing subaccounts are different from the parent and child account relationships that you can define on the **Account File Related Accounts** screen in PolicyCenter.

#### See also

- “Billing Accounts” on page 517 in the *Integration Guide*
- “Related Accounts Overview” on page 324

## Invoice Streams for an Account

Each account or subaccount can have multiple invoice streams. The integration retrieves invoice streams from the billing system for the currently selected account. You can create a new invoice stream for an account. Invoice streams specify:

- Periodicity – How often to send an invoice, such as twice a month, monthly, or every other month.
- Payment method – Specify a credit or debit card. Includes the card number, which is partially hidden in PolicyCenter. You can also wait for receipt of payment if PolicyCenter does not initiate the payment.
- Day of month – When to send the invoice. For a twice-monthly stream, you specify two values for day of the month.
- Due date or invoice date – Whether the day of month specifies the day that the invoice is due or the day to send the invoice.

- Automatic or manual – Whether to automatically charge or debit the payment from the payment method.

The billing system sends a single invoice for all policies with the same invoice stream. For example, if a personal auto policy and a homeowners policy have the same invoice stream, the insured receives a single invoice for both policies. You can use invoice streams for automatic payments.

## Policy Overview

Policies exist in both PolicyCenter and the billing system. A policy is associated with an account.

If you want to share a policy between PolicyCenter and the billing system, you must create the policy in PolicyCenter.

### Alternate Billing Accounts

In PolicyCenter, you can specify an alternate billing account for a policy. This account can be another account or a subaccount retrieved from the billing system. For example:

- **Subaccount** – A company has one subaccount for paying workers' compensation policies, and another subaccount for paying business auto premiums. The company can easily see how much they spend for each type of policy.
- **An arbitrary account** – An agent starts a personal auto submission for a young adult. The policy is in the parent's account. However, the young adult's uncle wants to pay for the policy. The agent sets the uncle's account as the alternate billing account. The uncle's account is not a subaccount of the parent's account.

### Invoice Stream for a Policy

For a given policy, PolicyCenter displays the invoice streams for the currently selected account or subaccount. You can only select an invoice stream that matches the currently selected installment plan. For example, if you have selected a monthly installment plan, you can only select monthly invoice streams.

#### See also

- “Invoice Streams for an Account” on page 703

## Sending Charges and Other Information to Billing

After establishing an account and policy period with billing and payment methods, the integration sends charges to the billing system.

For every policy transaction, PolicyCenter sends financial charges to the billing system. These are known as charges in BillingCenter. These charges include increases or decreases in premium, taxes, or fees.

When the user binds a policy transaction (job) that can generate premium transactions, PolicyCenter sends billing information, such as the installment plan and the invoice stream, to the billing system. Even if there are no charges, the billing system may need to know about a change to the policy period. For example, the billing system may need to know whether the change canceled or reinstated the policy period.

## Contact Overview

In PolicyCenter, each account must have one contact who is the account holder. Each account can have any number of billing and accounting contacts. Each policy period can have one billing contact. PolicyCenter sends all account holder and billing contacts to the billing system.

If you are integrated with BillingCenter, the account holder is the default primary payer in BillingCenter.

In the default integration, PolicyCenter propagates contacts to the billing system. This propagation occurs when you create new or update existing contacts in PolicyCenter. The integration sends the fields that are of interest to the billing system.

## Policy Transactions That Create a New Policy Period

PolicyCenter creates a new policy period for submission, rewrite, and renewal policy transactions (jobs). When you create a new policy period, PolicyCenter sends a message to the billing system to create an equivalent policy period.

For submission, rewrite, and renewal policy transactions, you enter billing information on the **Payment** screen. Some fields on policies and policy periods are only for use by the billing system. These fields include the payment method (direct bill, agency bill, or list bill). You set initial values for these fields in PolicyCenter so that you usually do not have additional setup in the billing system. After sending the policy or policy period to the billing system, you edit these fields in the billing system. For more information, see “Working with the Payments Screen in the Billing System Integration” on page 711.

The following illustration shows the steps and messaging that occur between the applications when you quote a policy for the following jobs: submission, renewal, and rewrite.

<b>PolicyCenter action</b>	<b>Messaging</b>	<b>Billing system action</b>
1. User quotes. 2. User advances to Payment screen.	3. PolicyCenter sends message to get billing options, installment plans, and invoicing plans.	4. Check to see if the producer code of record allows agency bill for this producer. 5. Look up installment and invoicing plans.
6. User selects billing method. 8. User selects installment plan. 9. (Optional) User clicks to preview payments that billing system returned.	6. Billing system returns billing options, available installment plans, and invoicing plans.	
7. User selects invoicing plan. 14. User binds policy.	10. PolicyCenter sends message with the selected payment plan.	11. Calculate payment schedule.
13. User selects invoicing plan.	12. Billing system returns payment schedule which can be viewed in PolicyCenter.	
15. If the billing system does not know about the account, PolicyCenter sends account information. 16. If the user adds an invoicing plan, PolicyCenter sends new invoicing information.		17. If account is unknown, create account. 18. If new invoicing plan, add to invoicing plans.
19. PolicyCenter sends message to create a new policy and/or policy period in billing system. In that same message, PolicyCenter sends billing charges related to that policy transaction.		20. Create policy or policy period on the account. 21. Process and apply the new charges

#### See also

- “Billing Flow for New-Period Jobs” on page 523 in the *Integration Guide*

## Policy Transactions That Create Midterm Changes

The issuance, policy change, cancellation, and reinstatement policy transactions (jobs) create midterm changes.

The following illustration shows the steps that occur between the applications when you quote a policy for job types issuance (not submission), policy change, cancellation, and reinstatement.

PolicyCenter	Messaging	Billing system
1. User quotes in PolicyCenter		
2. User advances to Payment screen.  <b>Note:</b> This step applies to most jobs, but not Cancellation.		
3. User binds policy.	4. PolicyCenter sends message to update the policy and/or policy period in the billing system. In the same message, PolicyCenter sends billing charges related to that job.	
		5. Billing system actually process and applies new charges.

The integration must handle certain midterm policy period changes and propagate the changes to PolicyCenter or the billing system.

You make certain mid-term changes in the billing system. These include:

- **Changes to the billing or payment method** – You view the billing or payment method in PolicyCenter. You make midterm changes to these in the billing system.
- **Changes to the producer of service or producer of record** – You change the producer of service or producer of record in PolicyCenter. If you want to give commission credit to the new producer in midterm, you must make this change in the billing system.

You make other midterm policy period changes in PolicyCenter. These include:

- **Changing policy period dates** – The integration pushes changes to the billing system.
- **Revised installments** – When you make a midterm policy change or reinstate a canceled policy, you cannot preview the revised payment schedule based on the new policy transaction. You can view the revised payment schedule after you make the change.
- **Moving a policy to a new account** – In PolicyCenter, you can move a policy from one account to another. You can also merge an account into another account, moving all policies to the new account. PolicyCenter sends notice of the changes to the billing system.
- **Holding return premiums when canceling with an audit pending** – PolicyCenter tells the billing system to hold the return premiums. PolicyCenter tells the billing system to release the hold when:
  - After completing the audit.
  - After removing a cancellation then reinstating the policy.
  - After canceling (again) the policy as a flat cancellation. There is no final audit because a full refund is automatic. (A flat cancellation cancels as of the beginning of the period with a full refund.)
  - After waiving the final audit.

### See also

- “Billing Implications of Midterm Changes” on page 526 in the *Integration Guide*

## Cancellations in the Billing System Integration

PolicyCenter always processes cancellations. However, the billing system can initiate a cancellation request that PolicyCenter cancel the policy.

### See also

- “Billing Implications for Cancellations and Reinstatements” on page 535 in the *Integration Guide*

### Cancellations That Start in PolicyCenter

Cancellations can start directly in PolicyCenter. When the transaction completes, PolicyCenter tells the billing system that it canceled the policy. The billing system then processes any return premiums that PolicyCenter sent.

Some examples of cancellations initiated by PolicyCenter are:

- For the purpose of doing a rewrite.
- At the request of the insured.
- When the carrier has grounds for cancellation. For example, the insured is found to have lied in their application or violated the contract.
- After binding a renewal, the insured informs the carrier that they are not going to take the renewal.

### Cancellations That Start in the Billing System

Some cancellations start in the billing system.

- **Cancel as soon as possible** – The primary example of this is cancellation for non-payment. If there are overdue invoices, the billing system starts a delinquency process. As part of this process, the billing system tells PolicyCenter to cancel as soon as possible. PolicyCenter calculates the actual cancellation date by using the minimum lead time required by law. When PolicyCenter actually cancels the policy, PolicyCenter sends a message to the billing system (just like any other cancellation). The BillingCenter integration uses this method.
- **Cancel on a specified date** – When integrating with a third-party billing system, the billing system controls the date of cancellation. The billing system must also keep track of notification lead times.

If the billing system receives payment before PolicyCenter cancels the policy, then the billing system sends a message to PolicyCenter to rescind the cancellation.

### Cancellation of Not Taken Renewals

Cancellations for non-payment often occur if the insured decides not to renew their policy. Instead of telling the carrier that they wish to cancel, the insured simply does not pay. In this case, the cancellation reason is not taken rather than non-payment.

At renewal time, PolicyCenter binds the renewal, and sends the bound policy period to the billing system. The billing system initiates a cancellation if it does not receive payment in a specified amount of time.

BillingCenter has a workflow for this delinquency process. If the customer does not pay within a specified time, BillingCenter begins a delinquency process for non-payment of a renewal. If BillingCenter does not receive payment, then BillingCenter sends PolicyCenter a message to cancel the policy with a not taken cancellation reason. PolicyCenter flat cancels the policy period as of the effective date of the policy period.

### See also

- “Renewals or Rewrites in the Billing System Integration” on page 709

## Reinstatements in the Billing System Integration

When the carrier receives payment after canceling a policy, the carrier may choose to reinstate the policy. The carrier's choice is usually an underwriting decision, and a user processes the reinstatement manually in PolicyCenter. PolicyCenter sends billing information to the billing system, which applies the payment to the reinstatement charges.

Some carriers, particularly those that offer high-volume personal lines policies, may want to automate the handling of these late payments. You can customize the integration to support automated reinstatements.

### See also

- “Billing Implications for Cancellations and Reinstatements” on page 535 in the *Integration Guide*

## Renewals or Rewrites in the Billing System Integration

The integration must handle changes related to renewals and rewrites.

### Account Creation for Conversion on Renewal

The *conversion on renewal* process moves policies into PolicyCenter at renewal time. In some cases, these policies are also new to the billing system. If PolicyCenter creates an account during a conversion on renewal, the integration checks to see if that account has already been created in the billing system. If it has not, PolicyCenter tells the billing system to create a new account, similar to what happens in a submission.

For more information on conversion on renewal, see “Policy Renewal Web Services” on page 179 in the *Integration Guide*.

### Copying Billing Fields to the new Period on Renewal or Rewrite

When PolicyCenter creates a new period for a renewal or a rewrite, it retrieves the billing and payment methods from the billing system. PolicyCenter retrieves these just in case the billing system changed them during the current period. PolicyCenter displays these fields on the payment page and allows the user to change these values. PolicyCenter copies other fields that the billing system maintains from the prior period to the new period.

### Renewal Process Flows

PolicyCenter supports the following renewal process flows:

- Bind and cancel
- Renewal offer
- Confirmed renewal

In the default configuration, PolicyCenter uses the *bind and cancel* renewal flow for all lines of business. When you bind a renewal, PolicyCenter sends charges to the billing system. PolicyCenter then does a flat cancel for reason **Policy not-taken** if the billing system receives no payment for that period. If partially paid, then PolicyCenter cancels for reason **Non-Payment**. See “Cancellation of Not Taken Renewals” on page 708 for details.

The default configuration contains the *renewal offer* flow which binds only after payment. You can configure this renewal flow for a particular line of business. Under this approach, you make the decision to renew or not renew, but instead of actually binding the renewal, you consider it a renewal offer. When you make the renewal offer, PolicyCenter sends a renewal notice (including pricing and payment plans). PolicyCenter does not send charges to the billing system (since no policy transaction completed). When the billing system receives payment, it sends a message to PolicyCenter to bind the renewal. If the billing system does not receive payment, the PolicyCenter renewal flow times out. PolicyCenter considers the renewal as not taken.

The default configuration contains the *confirmed renewal* flow which provides confirmation from the billing system that the insured completed payment. PolicyCenter knows if the policy was confirmed and is legally binding. The bind and cancel flow does not provide either of these.

**See also**

- “Multiple Choices of Renewal Flow” on page 529 in the *Integration Guide*

## Final Audits in the Billing System Integration

Auditors do not complete final audits until some time after the policy expiration date. Audits require special handling by the billing system.

Under normal circumstances, a billing system closes the policy period when there are no outstanding charges, no outstanding balance, the expiration date has passed, and all premium is earned. Basically, closing the period means that the billing system is not expecting any more activity on that period.

For policies that require a final audit PolicyCenter tells the billing system to hold the period open pending final audit.

When the auditor completes an audit, PolicyCenter sends a message to the billing system that contains incremental premium charges resulting from the audit.

**See also**

- “Billing Implications of Audits” on page 537 in the *Integration Guide*

## Reversing and Revising an Audit

An auditor can reverse or revise an audit.

If an auditor revises an audit, PolicyCenter sends the additional charges to the billing system.

If an auditor reverses an audit, PolicyCenter tells the billing system that there was an audit reversal and sends the offsetting charges to the billing system. The billing system must change the policy period so that it can accept a new audit. PolicyCenter also tells the billing system to keep the period open in expectation of a new audit.

## Premium Reporting in the Billing System Integration

When a user completes a premium report in PolicyCenter, the integration sends premium transactions as charges to the billing system. The user can enter a value for **Payment Received** when filling out the premium report.

PolicyCenter informs the billing system whether or not payment was received. This information allows the billing system to wait for payment to be posted to the billing system. It is common to enter the premium report before depositing the payment to avoid generating an invoice when the payment is already received.

The integration also handles the case in which the billing system receives the premium payment before PolicyCenter. The payment and the premium report are often sent together. The payment is often deposited before the report makes its way to the premium audit department for entry into PolicyCenter.

**See also**

- “Billing Implications for Premium Reporting” on page 538 in the *Integration Guide*

## Deposits

A deposit is collateral collected up front on a policy that will be otherwise billed based on reporting. PolicyCenter determines the deposit required based on total premium and the deposit percentage for the reporting plan chosen. PolicyCenter displays the deposit on the **Payment** page.

The deposit requirement is sent to the billing system as part of sending charges for each policy transaction. PolicyCenter sends the full deposit amount needed, rather than incremental changes to the deposit required. PolicyCenter expects the billing system to collect money for the deposit or release any extra being held. At the end of the period (when sending the final audit), the deposit amount is \$0. This amount normally causes the billing system to release it.

## Working with the Billing System Integration

This topic describes some of the changes in the user interface when you enable an integration with a billing system such as BillingCenter.

- “Working with the Payments Screen in the Billing System Integration” on page 711
- “Viewing the Policy Period in BillingCenter” on page 713
- “Working with Accounts from the Billing System” on page 713
- “Working with Policies in the Billing System Integration” on page 714

**Note:** For instructions on enabling the BillingCenter integration, see “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*.

### Working with the Payments Screen in the Billing System Integration

The **Payment** screen appears in a submission, renewal, or rewrite policy transaction.

You view the **Payment** screen after quoting but before binding the policy. PolicyCenter displays billing methods, payment methods, and associated installment or reporting plans retrieved from the billing system. After you select a payment method, you can preview payments retrieved from the billing system. This part of the wizard is similar for all lines of business.

You must specify a billing method, a payment plan for the policy, and the amount of deposit collected, if any.

The amount of payment collected by the agent is stored in PolicyCenter. It is not sent to the billing system. The billing system waits for the payment to arrive before starting the delinquency process.

This screen optionally integrates with a billing system through the currently registered billing system plugin. In the default configuration, the `StandAloneBillingSystemPlugin` provides sample billing system data. In a production system, you can use the plugin that connects to BillingCenter or write your own billing system plugin.

If PolicyCenter is configured as a multicurrency system, the **Payment** screen displays all amounts in the settlement currency.

#### See also

- “Enabling Integration between BillingCenter and PolicyCenter” on page 98 in the *Installation Guide*
- “Billing Integration” on page 509 in the *Integration Guide*

### Premium Summary

At the top of the screen, the **Premium Summary** shows the:

- **Total Premium**
- **Taxes and fees**
- **Total Cost** – The sum of the previous fields.

## Billing

Under the **Billing** heading, you can select:

- **Billing Method** – This drop-down list displays the billing methods retrieved from the billing system. In the default configuration, the choices are **Agency Bill**, **Direct Bill**, and **List Bill**. PolicyCenter sends a message to the billing system to see if the producer code of record for the producer allows agency bill. The billing system also returns a list of payment methods available for this type of policy and account.
- **Alt Billing Account** – Select an alternate billing account.
- **Billing Contact** – Select a person or company as the billing contact.

### Alternate Billing Account

Use the **Alt Billing Account** field to select an alternate billing account, if any. There are several reasons to use an alternate billing account:

- **List Bill** – For this billing method, you must select an alternate billing account because the policy premium is paid by an account that is not the current account. For example, on some mortgages, the mortgage company is responsible for paying the premium for a homeowners policy. The customer's mortgage payment includes the cost of the premium. List bill can also be used for an employee who uses his personal car for business. The employer pays the premium of his personal auto policy.

Select **Alt Billing Account** → **Search**. PolicyCenter displays the **Search Billing Accounts** screen. If you are connected to a billing system, you can search for list bill accounts on the external system. If you are connected to the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample billing accounts. If you are integrated with **BillingCenter**, the plugin returns a list of billing accounts retrieved from **BillingCenter**.

- **Direct Bill** – The alternate billing account is optional. An alternate billing account is useful in the following situation. An agent starts a personal auto submission for a young adult. The policy is in the parent's account. However, his uncle wants to pay for the policy. Therefore, the agent sets the uncle's account as the alternate billing account. The uncle's account is not a subaccount of the parent's account.

You can select **Search** to search for the billing account. The search finds accounts in PolicyCenter. These accounts have a corresponding account in the billing system. Select **Alt Billing Account** → **Billing Subaccounts** to select a subaccount of the current account. The subaccount exists in the billing system. By limiting the search to accounts in PolicyCenter, security controls the accounts that the user finds.

- **Agency Bill** – The user can select an **Installment Plan**. Billing is negotiated between the agent and the carrier. Therefore, the screen hides the **Alt Billing Account** selection. The user cannot choose an **Invoicing** stream.

### Billing Contact

The **Billing Contact** is the person to contact if there are questions about billing. For example, the billing contact can be the account holder, an accounts payable department, or the person in charge of billing at a company. This field appears if the **Billing Method** is **Direct Bill**.

## Schedule

Under the **Schedule** heading, you can select an installment or reporting plan and invoicing. The installment or reporting plans are retrieved from the billing system.

### Installment Plan

The **Installment Plan** lists installment plans which specify numbers of payments and a down payment percentage. The list of installment plans is retrieved from a billing system. If you are integrated with the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample installments plans. If you are integrated with **BillingCenter**, the plugin returns a list of valid installments plans for the selected list bill payer. The integration retrieves installments plans from **BillingCenter**.

Select an installment or reporting plan then click **Preview Payments** to see payment amounts and due dates generated by the billing system.

### Invoicing Stream

The **Invoicing** fields appear if you select a **Billing Method** of **Direct Bill** or **List Bill**. The installment plan and alternate billing account determine which invoice streams are available and which invoice stream is the default.

PolicyCenter displays all invoicing methods retrieved from the billing system. When integrated with the **StandAloneBillingSystemPlugin** or the default integration with BillingCenter, PolicyCenter displays radio buttons only for the methods that apply to the currently selected **Installment Plan**. For example, if you select a monthly installment plan, PolicyCenter hides the radio button in the **Select** column of an invoice plan that sends invoices every other month. The **StandAloneBillingSystemPlugin** simulates a carrier configuration where the carrier does not send out invoices when payment is monthly.

If the **Billing Method** is **List Bill**, you can select an existing invoice stream, but you cannot add a new invoice stream.

If the **Billing Method** is **Direct Bill**, you can select an existing invoice stream or add a new invoice stream. If you add an invoice stream, that invoice stream is saved to the billing system. Set **Invoice to Due Date** or **Invoice Date** to specify whether **Day of Month** specifies the day that the invoice is due or the day to send the invoice. Specify the **Day of Month**. Select **Automatic** or **Manual** payments.

Click the **Add** button to setup a new payment method in a payments system. In the default configuration, PolicyCenter displays a **Demo Payment System** screen to enter payment information for a credit card or ACH/EFT. If you integrate with a payments system, clicking the **Add** button could take the user to a similar screen that the payments system displays. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payments system, “Payment Integration” on page 553 in the *Integration Guide*.

If you select **Automatic** for **Payment**, payments are automatically charged or debited. You must select or add a **Payment Method**. The payment methods are **Credit Card** or **ACH/EFT**.

### Payments

In **Payments**, you can enter the amount of money **Collected by agent**. Use this field if the agent collects the down payment or deposit amount when the policy is bound. This field is often useful in personal lines of business.

PolicyCenter collects this information for direct bill policies so that the billing system can:

- Not send an invoice to the insured because the amount has already been collected by the agent.
- Do an *agency sweep* to collect the payment from the agent. This may be done by directly debiting an account specified by the agent for this purpose.

**Note:** The BillingCenter integration does not support this functionality.

## Viewing the Policy Period in BillingCenter

If the BillingCenter integration is enabled, you can view the policy period in BillingCenter after binding the policy.

1. Navigate to a policy, and click **Billing** in the left sidebar.
2. Click **View in BillingCenter**. This action takes you to the policy period summary in BillingCenter.
  - a. If you are not logged into BillingCenter, BillingCenter opens a new window to the login screen.
  - b. BillingCenter displays the policy.

In BillingCenter, you can view the payment schedule, transactions, charges, and commissions.

## Working with Accounts from the Billing System

In PolicyCenter, the **Billing** screen for an account displays account fields maintained by the billing system. For each account, you can view this page by clicking **Billing** in the left sidebar. This screen contains the following information:

- **Billing Account** – Displays the billing account. You can also display billing subaccounts of the current account. Use the drop-down list to select a subaccount and view the details of that account.
- **Account Balances** – Displays **Billed Outstanding**, **Unbilled**, and **Unapplied Funds**.
- **Collateral** – An additional asset or amount that may be required of an insured to secure coverage for a new or renewed policy. The insured can satisfy the collateral requirement with either cash, letters of credit (LOC), or a combination of both.
- **Primary Payer** – Displays the **Name**, **Address**, and **Phone** of the primary payer.

In the default configuration, PolicyCenter does not have a screen for specifying a billing subaccount. Billing subaccounts are retrieved from a billing system. In the default configuration, the `StandAloneBillingSystemPlugin` simulates retrieving billing subaccounts from a billing system. The large sample data set has examples of billing subaccounts. For information on how to load the sample data, see “[Installing Sample Data](#)” on page 52 in the *Installation Guide*.

**IMPORTANT** Billing subaccounts are different from the parent and child account relationships that you can define on the [Account File Related Accounts](#) screen.

The [View In BillingCenter](#) link allows you to view billing account details in BillingCenter. If you are logged into BillingCenter, the link jumps directly to the account. Otherwise, you go to a login screen. After logging in, BillingCenter displays the account.

In BillingCenter, you can view billing details. If you have sufficient permissions, you can start a delinquency or log a trouble ticket.

## Policy Terms Tab

The [Policy Terms](#) tab at the bottom of the screen displays summary information for individual policy terms.

The [Owned Policies](#) section displays policies owned by this account. The summary information includes the policy number, effective dates, billing method, alternate billing account, balances, and invoice stream for individual policy terms.

The [Other Policies Billed to this Account](#) section displays policies billed to this account but not owned by this account. The summary information includes the policy number, effective dates, owning account, balances, and invoice stream.

## Invoices Tab

The [Invoices](#) tab displays invoices retrieved from the billing system. You can choose to display invoices for the last three, six, and 12 months. For each invoice, the summary information includes statement and due dates, invoice number, invoicing period and payment instrument, status, and balances.

### See also

- “[Account File](#)” on page 323

## Working with Policies in the Billing System Integration

In PolicyCenter, the [Billing](#) page for a policy period displays account fields maintained by the billing system. For each account, you can view this page by clicking [Billing](#) in the left sidebar.

The [View In BillingCenter](#) link allows you to view the details in BillingCenter. If you are logged into BillingCenter, the link jumps directly to billing details for the policy period. Otherwise, you go to a login screen. After logging in, BillingCenter displays billing details for the policy period.

# Claim System Integration

The default configuration of PolicyCenter includes a completely functional integration with Guidewire ClaimCenter. You can also integrate PolicyCenter with the claim system of your choice. This topic describes how PolicyCenter integrates with a claim system in general and ClaimCenter in particular.

This topic includes:

- “Claim System Integration Overview” on page 715
- “Accessing Summary Loss Information from the Claim System” on page 716
- “Loss Claims Notification at Renewal” on page 718
- “Large Loss Notification” on page 719
- “Permissions for Working with Claims” on page 719
- “Claim Search Plugin” on page 720

**See also**

- “Enabling Integration between ClaimCenter and PolicyCenter” on page 95 in the *Installation Guide*
- “Claim and Policy Integration” on page 557 in the *Integration Guide*

## Claim System Integration Overview

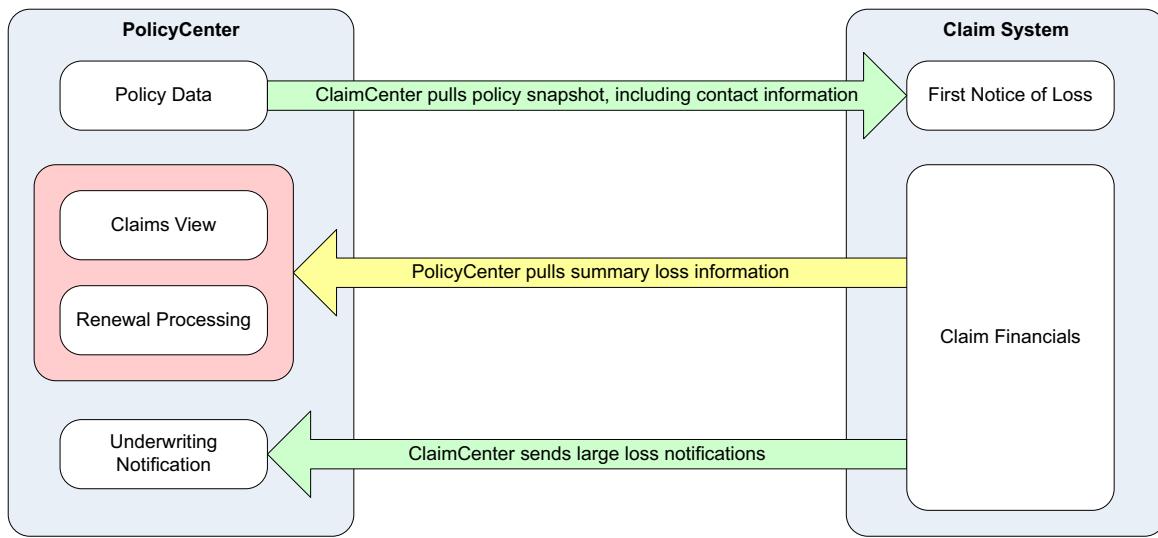
In the default configuration, PolicyCenter is set up to retrieve claims from a claim system. The default installation includes an integration with ClaimCenter.

The integration provides the following features:

- You can view claims on a policy or an account. The integration retrieves claims from the claim system and displays a summary.
- For all policy transactions, you can view claims on a policy.
- If you enable the ClaimCenter integration, the PolicyCenter user interface contains links that allow you to view claims in ClaimCenter.

- During renewal processing, underwriting automatically evaluate claim loss history during the renewal evaluation.

The following illustration shows the integration between PolicyCenter and a claim system at a high level.



#### See also

- ClaimCenter Application Guide* for information about the parts of the integration that originate in ClaimCenter. The parts of the integration that originate in ClaimCenter are:
  - ClaimCenter pulls policy snapshot
  - ClaimCenter sends large loss notification

## Accessing Summary Loss Information from the Claim System

In PolicyCenter, the underwriter can view loss history for accounts and policies. PolicyCenter retrieves the following loss summary information from the claim system:

- |                             |                                |
|-----------------------------|--------------------------------|
| • Claim number              | • Value for remaining reserves |
| • Loss date                 | • Value for total paid         |
| • Loss cause                | • Value for total recoveries   |
| • Status                    | • Value for net total incurred |
| • Description               |                                |
| • Information on litigation |                                |
| • Information on injuries   |                                |

When integrated with ClaimCenter, the user can view the claim in ClaimCenter.

## Viewing Loss Claims for Policies

For policies, you can view loss claims from the **Risk Analysis** section of all the policy transactions. Typical users are the underwriter and risk management staff who look for loss information from within a single policy. They usually look for this information while assessing whether to grant a renewal.

You can search for claims by loss date. If you select **Since**, your choices are **Any**, **Last 30 days**, **Last 90 days**, and **Last year**. If you select **From**, you can specify a date range to retrieve claims for all policy periods within that range. PolicyCenter sorts the claims by policy period.

The search also retrieves archived claims that fall within the search dates. In ClaimCenter, archived claims have only a limited amount of information about the claim. Therefore, archived claims do not display values for all fields in the claim summary list or **Claim Details**.

PolicyCenter displays only the policy periods that include a claim loss date. If there was no claim with a loss date during the policy period, that period does not display. The claim system returns only those policies for which the user has view access.

The **Total Incurred** column displays the cost of each claim. Archived claims do not display a value for this field. The last row in this column displays the total incurred for all claims in the summary list.

In a submission policy transaction, **Risk Analysis** is part of the wizard steps. The **Claim** tab allows you to search for claims in the following ways:

- **Search By Related Policy** allows you to select a policy on the account and retrieve claims on that policy.
- **Search By Loss Date** allows you to search for claims since a particular date or within a date range.

Click **Search** to display a list of claims retrieved from the claim system.

## Policy Period Filtering

On the **Risk Analysis** screen, you can filter the search results by making a selection from the **Policy Period** drop-down menu. Your choices are:

- **All** – Display all claims. This choice is the default selection.
- **No policy in force** – Display claims logged when coverage was not in force.
- **Policy periods within the search range** – Display claims logged in the selected policy period.

Claims logged on a loss date when coverage was not in force display **No policy in force** in the **PolicyPeriod** column.

In the ClaimCenter integration, the ClaimCenter policy period can get out of date. When you make a change to a policy, the policy information for a claim is not usually updated in ClaimCenter. For example, if after filing the claim, a backdated cancellation changes the policy period, the policy period in ClaimCenter is not updated. If the claim is no longer in the policy period, PolicyCenter lists the claim under **PolicyPeriod** as **No policy in force**. The **Claim Detail** tab displays the **Policy Period Start** and **Policy Period End** from PolicyCenter.

## Claim Details

Click a claim summary to view the **Claim Details** tab. This tab displays claim information retrieved from the claim system such as:

- **Claim Type**
- **Claim Number**
- **Policy Period Start**
- **Policy Period End**
- **Loss Date**
- **Loss Cause**
- **Status**
- **Description**
- **Litigation**
- **Injuries**
- **Remaining Reserves**
- **Paid to Date**
- **Total Incurred**
- **Recoveries**

## View Claim in Claim System

The **Claim Details** has a **View Claim in Claim System** button. This button appears if the ClaimCenter integration is enabled and you have the permissions to view claims in the system. This button opens a window to ClaimCenter.

You must have a user account in ClaimCenter. If you are not logged in, the login screen appears. When you are logged in, ClaimCenter displays the claim. If you have single sign-on, ClaimCenter opens directly to the claim summary screen.

**Note:** PolicyCenter does not display the **View Claim in Claim System** link for archived claims.

## Viewing Loss Claims from an Account

The **Claims** screen in an account lets you view claims associated with the account. You only see claims for which you have view permission on the policy. You can view this screen by navigating to an account then clicking **Claims** in the left sidebar. This screen is similar to the **Claims** screen for policies except that it retrieves claims for all policies in the account. The account **Claims** screen displays **Product** and **Policy Number** in the account claim list and **Claim Details**.

Like policies, you can filter the search results by policy period. You can also filter the results by product. The **Product** drop-down menu displays products for which the account has policies.

## Loss Claims Notification at Renewal

At renewal, PolicyCenter checks for underwriting issues. One of the checks is to search for claims from the claim system. By default, the search uses the start and end dates of the policy period currently in force.

After the claim search, PolicyCenter creates one or more *claims underwriting issues*. PolicyCenter attaches claims underwriting issues to the renewal policy transaction. Typically, PolicyCenter creates an underwriting issue for any occurrence of claims within the requested date range. In many cases, the claims underwriting issue is not severe enough to block automated renewal. Claim underwriting issues and *authority grants* (described below) determine whether an issue blocks renewal.

The following table describes claims underwriting issues.

Claims underwriting issue	Description
Claim total incurred	This issue appears if at least one claim is found in the requested date range. The message displays the value of the highest cost claim returned. This issue blocks bind.
Ratio of claims total incurred to a policy written premium	This issue appears if the search returns at least one claim and the policy has a written premium for the in-force policy period. The message displays the sum of total incurred values for all returned claims divided by the written premium. This issue blocks bind.
Incidence of claims	This issue appears if the search returns at least one claim and the policy has a written premium for the in-force policy period. The message displays the number of claims returned from search divided by the written premium. This issue blocks bind.
Manual claim review needed	This issue appears if the search matches more claims than the system is configured to retrieve. This issue blocks bind.
Unable to retrieve claims information	This issue appears if the search was unable to retrieve claims information. This issue blocks bind.

*Authority grants* determine which claims underwriting issues block automated renewal. Claims underwriting issues require underwriter approval if the issue triggers an authority grant and the issue blocks bind.

The following table lists the authority grants of specific users.

User	Authority Grants
Renewal workflow	<ul style="list-style-type: none"> <li>• Claim total incurred &gt; \$10,000</li> <li>• Ratio of claims total incurred to policy written premium &gt; .5 (.5 = 50%)</li> <li>• Incidence of claims &gt; .0005 (5 claims / \$10,000 of premium)</li> </ul>
Underwriter 1 & Underwriter 2	<ul style="list-style-type: none"> <li>• Claim total incurred &gt; \$20,000</li> <li>• Ratio of claims total incurred to policy written premium &gt; .90 (.90 = 90%)</li> <li>• Incidence of claims &gt; .0010 (10 claims / \$10,000 of premium)</li> <li>• Manual claim review needed</li> <li>• Unable to retrieve claim information</li> </ul>
Underwriter Supervisor	<ul style="list-style-type: none"> <li>• Claim total incurred: no limit</li> <li>• Ratio of claims total incurred to policy written premium: no limit</li> <li>• Incidence of claims: no limit</li> <li>• Manual claim review needed</li> <li>• Unable to retrieve claim information</li> </ul>

#### See also

- “Underwriting Authority” on page 411

## Approvals

If a claim underwriting issue blocks during automated renewal, PolicyCenter creates an activity for the underwriter to review the issue and suspends automated processing. To avoid the need for additional review, the underwriter can grant approval for a higher amount than the current value. When all issues are cleared through issuance, the underwriter has two choices. The underwriter may then select to renew the policy manually or place the policy back into the automated renewal flow.

## Large Loss Notification

ClaimCenter sends a large loss notification to PolicyCenter when the gross total incurred crosses a threshold in ClaimCenter. This threshold is configurable. When the threshold is crossed, PolicyCenter creates a `UWReferralReason` on the policy and an activity for the underwriter assigned to the policy.

For details, see the *ClaimCenter Application Guide*.

## Permissions for Working with Claims

If the user has permission to see the risk analysis page, they have sufficient permission to access the `Claims` tab and search for claims.

There are two permissions that allow the user to view claims in ClaimCenter.

- **View claim system** – The user sees the **View Claim in Claim System** button. The code for this permission is `viewclaimsystem`.
- **View restricted claim** – The user can view restricted fields on restricted claims. The code for this permission is `viewrestrictedclaim`.

In the default application, users with underwriter role have the **View claim system** permission. Users with the underwriter supervisor role have the **View claim system** and **View restricted claim** permission.

## Restricted Fields in Claims

ClaimCenter restricts claims of various types. ClaimCenter can restrict claims by permission controls and with rules about who can view the claims. These rules are Access Profiles, which the Access Control List defines.

PolicyCenter maps all ClaimCenter restricted claim types to *restricted claim*. In PolicyCenter, some of the fields in the claim detail for restricted claim are hidden if user does not have the **View restricted claim** permission. You can modify your ClaimCenter configuration to control which fields are considered restricted. This configuration affects which users can view the fields in PolicyCenter.

For restricted claims, you may want to hide restricted fields in the user interface. In the default application, the fields for litigation and injuries do not display for restricted claims. You can customize the application to display or hide fields by setting the visibility attribute of the field in the `ClaimDetailsDV` PCF file:

```
visible=ClaimDetail.isClaimDataAvailable()
```

## Claim Search Plugin

The integration with ClaimCenter is handled by the `GWClaimSearchPlugin`. This plugin calls the ClaimCenter `PCClaimSearchIntegrationAPI` and translates the result into PolicyCenter claim objects. The plugin has methods that do the following:

- Search for claims
- Retrieve details of an individual claim
- Grant a user view permissions on a claim

To learn more about this plugin, see “Claim Search from PolicyCenter” on page 557 in the *Integration Guide*.

# Contact Management System Integration

A contact management system maintains contacts in a central location. These contacts can be shared across applications. The default configuration of PolicyCenter includes an integration with Guidewire ContactManager. You can also integrate PolicyCenter with the contact management system of your choice. In the default configuration, the integration with ContactManager is not enabled.

This topic describes how PolicyCenter integrates with a contact management system in general, and ContactManager in particular.

You can integrate PolicyCenter with more than one contact management system. ContactManager can be one of these systems.

PolicyCenter uses contacts in accounts and policies in a various ways. Contacts represent named insureds, account holders, billing contacts, additional interests, and other roles on accounts and policies. PolicyCenter can store the contacts in its internal address book and in a contact management system. In this case, the contact management system is usually the system of record for contacts.

You can configure the integration to store part of the contact information in the contact management system and other parts in the PolicyCenter internal address book. Each application is the system of record for a portion of the contact information. For example, the contact management system is the system of record for basic contact information such as name, address, and phone number. The PolicyCenter internal address book is the system of record for information related to the roles the contact plays on the account or policy.

This topic includes:

- “Contact Management System Integration Overview” on page 722
- “Searching for Contacts Within a Contact Management System” on page 722
- “New and Updated Contacts” on page 723
- “Detecting Duplicates in the Contact Management System” on page 725
- “Duplicate Contacts in PolicyCenter” on page 728

- “Deleting, Removing, and Inactivating a Contact” on page 729
- “Customizing the Contact Management System Integration” on page 729

**See also**

- “Contact Integration” on page 575 in the *Integration Guide*
- “Integrating ContactManager with Guidewire Core Applications” on page 43 in the *Contact Management Guide*

## Contact Management System Integration Overview

In the default integration with a contact management system or with ContactManager, PolicyCenter is the primary user interface. The contact management system is the central repository for contact information.

### Creating a New Account

The user logs in to PolicyCenter and searches for an account. If the account is not found, PolicyCenter displays a menu item that enables the user to search for a contact from the contact management system. The user creates a new account with this contact as the primary account holder. The new account exists only in PolicyCenter. The contact management system does not store accounts.

### Adding Contacts to an Account or Policy

The user searches for contacts in the contact management system. The user can add these contacts to an account or policy.

If the contact does not exist in the contact management system, the user creates a new contact on an account or policy in PolicyCenter. PolicyCenter pushes a new contact to the contact management system when that contact is associated with an account that has a bound policy. For more information, see “Creating a New Contact in PolicyCenter” on page 724.

### Updating Contacts

The contact management system is the system of record for contact information. PolicyCenter pushes updates in real time to the contact management system.

When a user accesses an account in PolicyCenter, PolicyCenter uses the contact information stored in its internal database. When PolicyCenter pushes contact updates to the contact management system, the contact management system resolves the differences and pushes changes to all applications.

**See also**

- “New and Updated Contacts” on page 723
- “Pushing New and Updated Contacts” on page 723

## Searching for Contacts Within a Contact Management System

In PolicyCenter, you can search for contacts in the address book. If PolicyCenter is not integrated with a contact management system, PolicyCenter searches the internal address book. If PolicyCenter is integrated with a contact management system, PolicyCenter searches for contacts stored in the internal address book and the contact management system.

When you select **From Address Book** in PolicyCenter, you can select contacts from the internal address book and from the contact management system.

If the contact is currently in the contact management system, then PolicyCenter pulls the contact into its internal address book when you **Select** it.

In PolicyCenter, you can search for contacts from a contact management system in the following places:

- **Contact** tab → **Search** menu item.
- **Search** tab → **Contact** menu item.
- **Account File Contacts** screen – You can choose to create a new contact from the address book.
- **Policy job** screens – You can choose to create a new contact from the address book.

If PolicyCenter is integrated with ContactManager, the ContactManager search is limited to a contact type of **Person** or **Company** with the **Client** tag. In the ContactManager object model, these are the **ABPerson** and **ABCompany** subtypes of **ABContact**.

#### See also

- “Adding a Contact From the Address Book” on page 383
- “Searching for Contacts” on page 77 in the *Contact Management Guide*

## New and Updated Contacts

When PolicyCenter is integrated with a contact management system, you can create and update contacts in either system. The integration pushes contact changes to the other system.

### Pushing New and Updated Contacts

You can push new and updated contacts from PolicyCenter to a contact management system. You can configure when to push new or updated contacts to a contact management system. You can also push updates to specific fields rather than updating the entire contact.

Going in the opposite direction, you can push new contacts and updates to an existing contact from a contact management system to the associated contact in PolicyCenter. You can push updates to specific fields rather than updating the entire contact.

#### Pushing New and Updated Contact to ContactManager

In the default integration with ContactManager, you can make contact updates on both sides of the integration:

- You can make contact updates in PolicyCenter and then push the updates to ContactManager.
- You can make contact updates in ContactManager or in other applications. The integration then pushes updates in other applications to ContactManager. ContactManager broadcasts those updates to the other applications. In the default integration with PolicyCenter, ContactManager is the system of record for contacts. Therefore, PolicyCenter takes all updates coming from ContactManager.

Although PolicyCenter takes all updates from ContactManager, PolicyCenter may take a different action on the contact. For example, ContactManager deletes a policy address either directly as a contact update or when two contacts are merged. ContactManager pushes that update to PolicyCenter. However, PolicyCenter does not delete that address because it is in use on a policy. PolicyCenter removes the link that connects the contact with ContactManager by removing the **AddressBookUID**.

- PolicyCenter does not delete a contact that is in use.

In PolicyCenter, an update made to a policy contact might not be immediately pushed to the **Contact** entity. In PolicyCenter, the **Contact** entity is central contact record. PolicyCenter pushes the update to ContactManager only after updating the **Contact** entity. For example, PolicyCenter does not push a new contact to ContactManager until that contact is associated with an account that has a bound policy. See “Adding a Contact from the Contact Management System” on page 724 for more information. In addition, changes to a policy

contact might not be immediately synchronized with the account contact as described in “Revisioning Contact Information in Policies” on page 371.

All updates are asynchronous. Therefore, there can be conflicts when ContactManager and PolicyCenter update contact information at approximately the same time. Since ContactManager is the system of record for contact information, the ContactManager updates have precedence. If ContactManager cannot make a PolicyCenter contact update, ContactManager notifies PolicyCenter. PolicyCenter then creates an activity for a user to reconcile the change to the contact. The activity text describes the updates that were not made.

**See also**

- “Linking and Synchronizing Contacts” on page 169 in the *Contact Management Guide*

## Adding a Contact from the Contact Management System

When PolicyCenter retrieves a contact from the contact management system, PolicyCenter copies the contact to its internal address book. If PolicyCenter is integrated with ContactManager, this contact has the same AddressBookUID as the ContactManager contact. If PolicyCenter is integrated with another contact management system, you can configure PolicyCenter to set the AddressBookUID to the unique identifier from the contact management system. When a contact has the same AddressBookUID in both systems, the two contacts are linked. The integration copies a change to contact data in either application to the other application.

For example, in PolicyCenter you add a driver to a personal auto policy. On the **Drivers** screen, you choose **Add → From Address Book** and enter a last name. PolicyCenter searches for matches in its internal address book and the contact management system. Select a contact. If the contact is only in the contact management system, then PolicyCenter copies the contact to its internal address book.

The internal contact now links to a contact in the contact management system. The integration propagates a change to the contact in either the contact management system or PolicyCenter to the other system.

**See also**

- “Adding a Contact From the Address Book” on page 383

## Creating a New Contact in PolicyCenter

If a new customer calls to add a new driver to a personal auto policy, the agent creates a new contact for the customer.

The default integration does not immediately push all newly created contacts to the contact management system. When you create a new contact in PolicyCenter, the integration pushes that contact to the contact management system after that contact is associated with an account with a bound policy. This process avoids cluttering the contact management system with prospective customers. The prospective customer remains in the PolicyCenter internal address book and can be used again if the customer calls back.

You can also check for duplicates to make sure that the contact is not already in the contact management system.

In the base configuration, if PolicyCenter finds duplicate contacts, you can select one. The selected contact replaces the new contact, and the integration links that contact to the contact management system regardless of the state of the policies on the account. Any contact information for the new contact is overwritten. Alternately, you may decide that this is not a duplicate and add the new contact.

For example, a prospective customer calls to obtain a personal auto quote. The customer service agent creates a new contact with a new account. Then the agent starts a personal auto submission on the account. The agent quotes the policy. The customer is not ready to buy the policy, and says she will call back later. Any newly created contacts associated with the account remain only in the PolicyCenter internal address book. A week later, the customer calls back and says she wants to purchase the policy. The agent binds the policy, and the integration pushes the contacts associated with the account to the contact management system.

**See also**

- “Creating a New Contact on the Contact Tab” on page 375

### Information Required for Saving in ContactManager

In the default PolicyCenter user interface, you must enter name and primary address when entering a contact. If you modify this behavior, be aware that ContactManager requires certain minimum information to create a contact.

ContactManager requires the following fields for a contact that is a person:

- **First Name** and **Last Name**
- A primary address with the following fields:
  - **Address Line 1**
  - **City**
  - **State**
  - **ZIP Code**
  - **Address Type**

ContactManager requires the following fields for a contact that is a company:

- **Name** of company
- Tax ID or primary address as described above

### Information Required for Selecting a Contact Management System Contact in PolicyCenter

In the base configuration, an external contact must have certain fields to be selectable in PolicyCenter. To Select an external contact, PolicyCenter requires the following contact information:

- **First name** and **Last name** or company **Name**
- **Primary Address** – You must specify **Address Line 1**, **City**, **State**, **Postal Code**, and **Address Type**.

If the contact does not have the required information, then PolicyCenter displays the contact, but you cannot select it.

In the default integration with ContactManager, a contact must have the **Client** tag to be visible in PolicyCenter. A contact in ContactManager that does not have the **Client** tag is not returned in search results and does not appear in PolicyCenter.

## Detecting Duplicates in the Contact Management System

When you add a contact in PolicyCenter, PolicyCenter checks for duplicates in ContactManager or the contact management system. The contact management system must support checking for duplicates.

**Note:** PolicyCenter does not check for duplicates in its internal address book.

**See also**

- “Creating a New Contact on the Contact Tab” on page 375 for information about the **Check for Duplicates** button.

### Detecting Duplicates When Integrated with ContactManager

When ContactManager checks for duplicate contacts, there are three types of matches:

- Exact match
- Potential match
- No match

This topic describes how ContactManager detects duplicates in the base configuration. You can customize this functionality. For example, you can change the fields that duplicate detection matches on, or change the matching logic.

#### See also

- “Understanding Find Duplicates Behavior” on page 171 in the *Contact Management Guide*

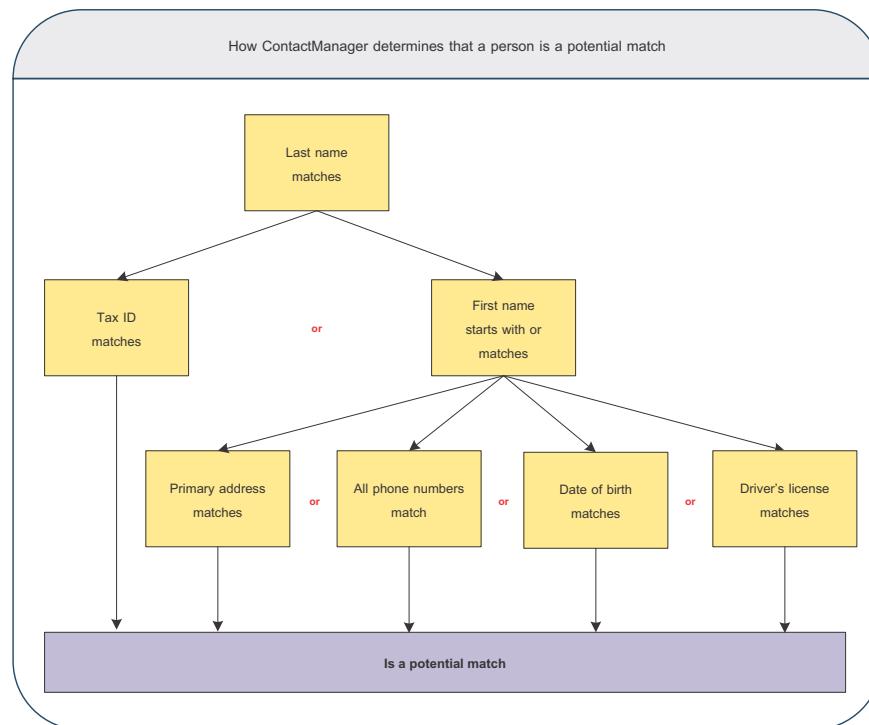
### How ContactManager Determines Duplicates for a Person

When checking for a duplicate person in PolicyCenter, you must enter the first name and at least one of the following:

- Primary address
- Phone number
- Date of birth
- Driver’s license
- Tax ID

In the base configuration, ContactManager first determines if the person is a potential match. If the person is a potential match, then ContactManager determines whether the person is an exact match.

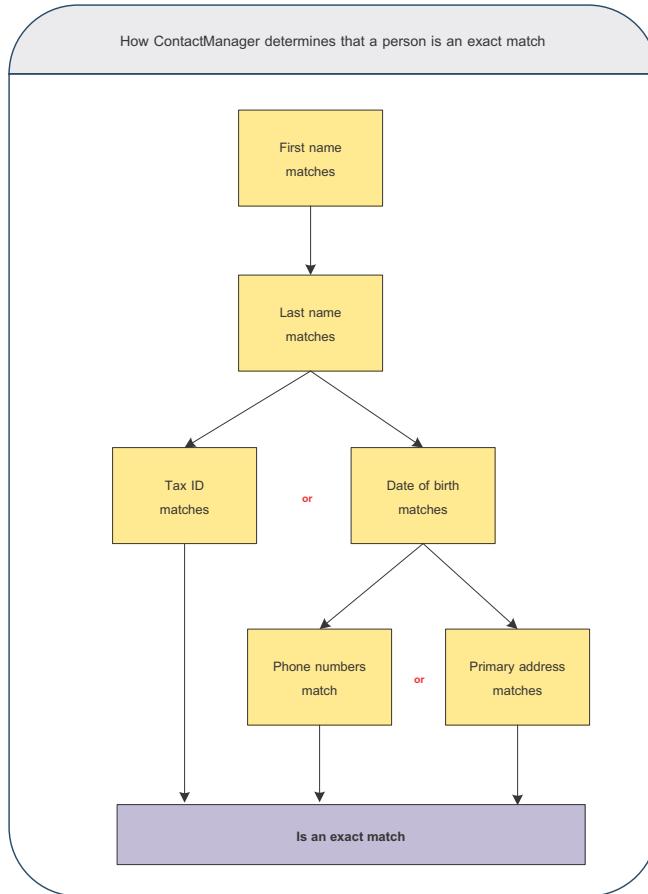
ContactManager first looks for a potential match. The following illustration shows the various paths through which ContactManager determines that a person is a potential match.



ContactManager uses the following criteria to determine if each of the following matches:

- Phone numbers – The phone numbers match if the numbers for home, work, cell, and fax match.
- Primary address – The primary address matches if Address line 1, state, city, and ZIP code match.
- License – The license matches if license number and license state match.

If a person is a potential match, then ContactManager determines if that person is also an exact match. The following illustration shows how ContactManager determines if a person is an exact match in the base configuration.

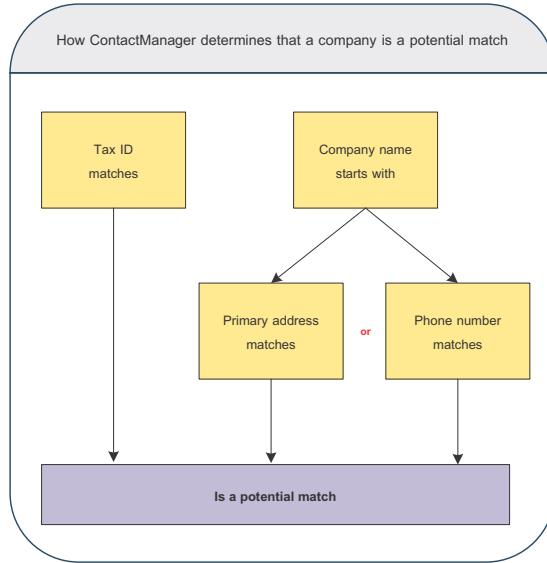


### How ContactManager Determines Duplicates for a Company

When checking for a duplicate company in PolicyCenter, you must enter the company name and at least one of the following: tax ID, primary address, or phone number.

ContactManager first determines if the company is a potential match. If the company is a potential match, then ContactManager determines whether the company is also an exact match.

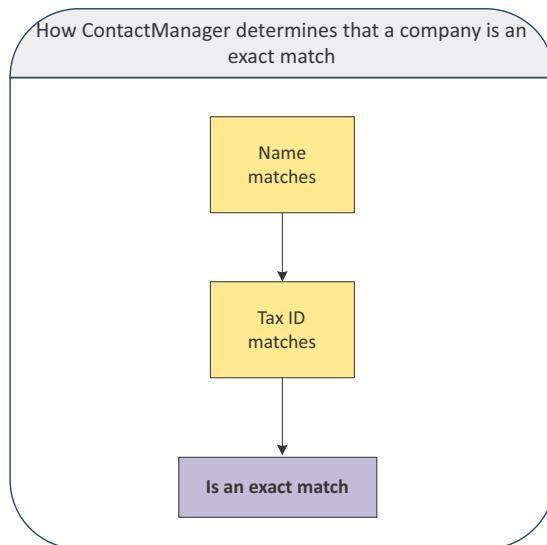
ContactManager finds a potential match if it finds a potential match through any path. The following illustration shows how ContactManager determines that a company is a potential match.



ContactManager uses the following criteria to determine if each of the following matches:

- Phone numbers – The phone numbers match if the numbers for home, work, and fax match.
- Primary address – The primary address matches if Address line 1, state, city, and ZIP code match.

If a company is a potential match, then ContactManager determines if that company is also an exact match. The following illustration shows how ContactManager determines if a company is an exact match in the base configuration.



## Duplicate Contacts in PolicyCenter

If the contact management system finds duplicate contacts, the contact management system sends a message to the other applications. PolicyCenter merges the duplicates into a single contact.

This topic describes the default integration between PolicyCenter and ContactManager or another contact management system.

When PolicyCenter merges contacts, it identifies one of the contacts as the *surviving* contact and the other as the *merged* contact. PolicyCenter might have both contacts, or only the surviving or merged contact in its internal database. After merging, PolicyCenter deletes the merged contact, if any.

PolicyCenter merges two contacts in the following ways:

- If contact information differs between duplicate contacts, the contact management system determines which is the surviving contact. The contact management system also controls which information, if any, to copy from the merged contact. The contact management system sends the new contact information to PolicyCenter.
- If both contacts exist on the same account and have overlapping account contact roles, PolicyCenter resolves information specific to the account contact role in favor of the surviving contact.
- If both contacts exist on the same policy, in the same role, PolicyCenter resolves role specific information in favor of the surviving contact.
- The contact management system merges all addresses on the duplicate contacts. The surviving contact contains these addresses.
- Addresses must be merged to the surviving contact. This merging can be done through an API or in the PolicyCenter user interface.

To merge contacts, use the ContactAPI methods `mergeContactAddressesByPublicId` and `mergeContactsByPublicId`. For more information, see “Contact Web Service APIs” on page 584 in the *Integration Guide*.

## Deleting, Removing, and Inactivating a Contact

In ContactManager, you can delete a contact, but you cannot deactivate a contact.

If you attempt to delete a contact in ContactManager, ContactManager checks with the other applications to see if that contact can be deleted. ContactManager deletes the contact only if all applications report that the contact can be deleted.

In the base configuration, you can remove a contact from an account if no policy or work order uses the contact, currently or on a previously bound policy transaction. You can also deactivate a contact on an account, and the contact remains on previously bound policies or policy transactions. However, you cannot add an inactive contact to new, changed, reinstated, rewritten, issued, or renewed policies. You can remove a contact on a policy. Removing or inactivating a contact on an account or policy does not delete the contact from PolicyCenter.

You can configure PolicyCenter to delete a contact. If the ContactManager integration is enabled and you delete a contact in PolicyCenter, the default integration does not send a delete message to ContactManager.

## Customizing the Contact Management System Integration

This topic describes some of the ways that you can customize the integration with a contact management system. You can also do these customizations to the integration with ContactManager.

PolicyCenter provides a set of integration points for integrating with a contact management system. These integration points support a wide variety of ways to integrate with contact management systems.

At a high level, PolicyCenter provides the following integration points:

- Add contacts from contact management system.
- Search for contacts from a contact management system.
- Push new contacts from PolicyCenter to a contact management system.
- Push or pull updates to contacts from a contact management system to PolicyCenter.

- Push or pull updates to contacts from PolicyCenter to a contact management system.
- View all places that use a particular contact.
- List policies, accounts, and work orders associated with a contact.
- Merge contacts in PolicyCenter that the contact management system identifies as duplicates.

## Entry Points into PolicyCenter

Entry points enable an external application to open PolicyCenter on a particular screen. An external application, such as a contact management system, can use the following entry points to PolicyCenter:

- **Account** – Given an account number or account ID, open that account directly from an external system.
- **Policy** – Given a policy number, open that policy file from an external system.
- **Work order** – Given a work order number, open that work order directly from an external system.

For more information, see “Contact Integration” on page 575 in the *Integration Guide*.

## Integrating with Multiple Contact Management Systems

You can integrate PolicyCenter with multiple contact management systems. ContactManager can be one of these systems. A carrier can use different systems to store different types of contact information. For example, a carrier might use one system for customer information such as named insured and drivers. The carrier might use another contact management system for additional interests, additional insureds, and labor contractors.

If you integrate with more than one contact management system, each contact must exist in one and only one external system. PolicyCenter does not support the same contact existing in more than one contact management system.