



Guidewire PolicyCenter™

Application Guide

Release 10.0.1

©2019 Guidewire Software, Inc.

For information about Guidewire trademarks, visit <http://guidewire.com/legal-notices>.

Guidewire Proprietary & Confidential — DO NOT DISTRIBUTE

Product Name: Guidewire PolicyCenter

Product Release: 10.0.1

Document Name: Application Guide

Document Revision: 25-February-2019

Contents

Guidewire InsurancePlatform™	27
About PolicyCenter documentation	27
Conventions in this document	29
Support	29

Part 1 Introduction to PolicyCenter

1 PolicyCenter overview	33
Product model	34
Lines of business in the base application	34
The policy lifecycle	35
PolicyCenter integration points	36
Integration with other Guidewire applications	37
PolicyCenter users	37

Part 2 PolicyCenter user interface

2 Navigating PolicyCenter	41
Logging into PolicyCenter	41
PolicyCenter login requirements	41
Log in to PolicyCenter	41
Setting preferences	42
Changing interface settings	42
Change the visual theme	44
Data entry support for input fields	44
Using the currency macro in currency fields	44
As-you-type formatting support for input fields	45
Highlight changed values	46
Selecting language and regional formats in PolicyCenter	46
Options for setting the display language	47
Options for setting regional formats	47
My Summary	47
Common areas in the PolicyCenter user interface	48
PolicyCenter tabs	49
Desktop tab	49
Account tab	53
Policy tab	53
Contact tab	53
Reinsurance tab	54
Search tab	54
Team tab	55
Administration tab	55
3 Changing the screen layout	57
Adjusting list views	57

Change list view column order	57
Sort list views	57
Hide and show columns in a list view	58
Grouping rows in a list view	58
Change the sidebar width	59
Managing layout preferences	59
Storage of layout preferences	59
Clear layout preferences	60
4 QuickJump	61
QuickJump overview	61
Using QuickJump	61
Chaining QuickJump destinations together	62
QuickJump behavior in wizards	62
Configuring QuickJump	62
QuickJump reference	62
Static items	62
Super user items	63
Account file items	63
Policy file items	64
5 Basic search	67
Basic search overview	67
Indexing free-text search data	67
Query and filter basic search fields	68
Exact or inexact basic search and ranking	68
Basic search results for fields with multiple matches	68
Basic policy search overview	69
Basic search user interface	69
Policy basic search screen	69
Working with basic search	72
Use basic policy search	72
Basic search index updates automatically	72
6 Advanced search	75
Advanced search overview	75
Minimum search requirements for advanced search	76
Advanced search for policies	76
Search contacts	77
Working with the Advanced Search tab	77
Assign a user to one or more policies or open policy transactions	77
Search quotes user interface	78
7 Saving your work	79
Unsaved work list	79
Saving your work in PolicyCenter wizards	79

Part 3

PolicyCenter policy transactions

8 Policy transactions	83
Overview of policy transactions	83
Key features of policy transactions	85
Preempting policy transactions	86
Out-of-sequence changes and policy transactions	86

Configuring policy transactions	87
9 Submission policy transaction	89
Submission overview	89
Understanding name clearance and risk reservation	90
The difference between quick quote and full application	90
Answering pre-qualification questions in a submission	90
Selecting an underwriting company in a submission	91
Reasons for underwriter review in submissions	92
Closing a submission	92
Differences between binding and issuing a policy in a submission	92
Expiring submissions	92
Copying submission information	93
Submission manager	93
Submission general steps	93
Working with submissions	94
Places to create a submission	95
Create a submission	95
Copy a submission	96
10 Issuance policy transaction	97
Issuance overview	97
Issuance general steps	98
Working with issuance	98
Issue a policy	98
11 Renewal policy transaction	101
Renewal overview	101
Starting renewals manually in the user interface	102
Starting renewals using the Policy Renewal web service	102
Starting renewals using a batch process	102
Renewal flows	103
Pre-renewal directions	103
Referral reasons on renewals	104
Underwriting issues in renewals	104
Policy changes and renewals	105
Renewal general steps	105
Working with renewals	107
Create a manual renewal	107
Create a renewal from a batch process	108
View your renewals	109
Create multiple versions of a renewal	109
Create a pre-renewal direction	109
View, edit, or delete an existing pre-renewal direction	110
12 Cancellation policy transaction	111
Cancellation overview	111
Scheduling a cancellation versus cancel now	112
Default cancellation effective date	112
Changing a cancellation	113
Rescinded versus withdrawn cancellations	114
Cancellation with pending renewals	114
Cancellation general steps	115
Working with cancellations	115
Cancel a policy	115
Rescind a cancellation	116

Change the cancellation effective date in a quoted cancellation	116
Change the cancellation effective date in a scheduled cancellation	117
13 Policy change transaction	119
Policy change overview	119
Using the Policy Review screen to verify changes	120
Editing the effective date of a policy change	120
Handling out-of-sequence policy transactions in a policy change	123
Using preemption in a policy change	123
Policy change general steps	123
Working with policy changes	124
Create policy change	125
Edit the policy change effective date	125
Change the policy expiration date in a policy change	126
14 Reinstatement policy transactions	127
Reinstatement overview	127
Reinstatement general steps	127
Working with reinstatements	128
Reinstate a policy	128
15 Rewrite policy transactions	129
Rewrite overview	129
Rewrite general steps	130
Working with rewrites	130
Rewrite a policy	131
Approve, request changes, or decline a rewrite	131
View rewritten policies	131
16 Rewrite new account policy transaction	133
Rewrite new account overview	133
Restrictions to rewrite new account started on source policy	134
Restrictions to rewrite new account started on target policy	137
Rewrite new account and the Policy Renewal Start batch process	137
Working with rewrite new account	138
Rewrite policy to new account	138
17 Premium audit policy transaction	139
Final audit overview	140
Where does the insurer send premium audit information?	141
Reversing and revising final audits	141
Adding a final audit	141
Premium report overview	142
Premium report audit schedules	143
Premium report trend analysis	143
Adding a premium report	143
Premium audit roles	143
Premium audit and other policy transactions	144
Premium audit general steps	146
Final audit general steps	146
Premium reports general steps	146
Working with final audits	147
View final audit schedule	148
View audit activities	148
Edit, start, or waive a final audit	148
Add a new audit	149

Enter audit data and complete final audit	149
Revise a final audit	150
Working with premium reports	150
View a premium report	151
View premium report activities	151
Select premium reports	151
Create activities for overdue premium reports	151
Enter premium report data	152
Edit or waive a premium report	152
Add a new premium report	152
18 Side-by-side quoting	153
Side-by-side quoting overview	153
Side-by-side quoting versus multi-version quoting	153
Side-by-side availability overview	154
Base data overview in side-by-side quoting	154
Side-by-side data overview	154
Side-by-side quoting process flow	155
Side-by-side quoting in the user interface	156
Tools menu items for side-by-side quoting	156
Side-by-Side Quoting screen	156
Policy Versions screen	158
Working with side-by-side quoting	158
Select side-by-side quoting in a submission	159
Edit a version in a policy transaction with side-by-side quoting	159
Bind and issue a side-by-side submission	160
19 Multi-version quoting	161
Multi-version quoting versus side-by-side quoting	161
Working with multi-version quoting	161
Create and compare multi-version quotes	161
Setting the maximum number of multi-version quotes	162

Part 4

Lines of business

20 Line of business overview	165
Additional lines of business	165
Developing a new line of business	166
Affinity groups	166
21 Businessowners	167
Businessowners screens	167
Offerings screen for businessowners	167
Qualification screen for businessowners	167
Policy Info screen for businessowners	168
Businessowners line screen	169
Locations screen for businessowners	170
Buildings screen for businessowners	171
Modifiers screen for businessowners	172
Risk Analysis screen for businessowners	172
Policy Review screen for businessowners	172
Quote screen for businessowners	172
Forms screen for businessowners	173
Payment screen for businessowners	173

Businessowners object model	173
Businessowners object model overview	173
Coverages in businessowners	174
Modifiers in businessowners	175
22 Commercial auto	177
Commercial auto screens	177
Offerings screen for commercial auto	177
Qualification screen for commercial auto	178
Policy Info screen for commercial auto	178
Commercial auto line screen for commercial auto	180
Locations screen for commercial auto	181
Vehicles screen for commercial auto	181
State Info screen for commercial auto	183
Drivers screen for commercial auto	184
Covered Vehicles screen for commercial auto	184
Modifiers screen for commercial auto	184
Risk Analysis screen for commercial auto	185
Policy Review screen for commercial auto	185
Quote screen for commercial auto	185
Forms screen for commercial auto	186
Payment screen for commercial auto	186
Commercial auto object model	186
Commercial auto object model overview	186
Coverages in commercial auto	187
Modifiers in commercial auto	188
Locations in commercial auto	188
Drivers in commercial auto	188
Jurisdictions in commercial auto	188
23 Commercial package policy	189
Commercial package screens	189
Offerings screen for commercial package	189
Qualification screen for commercial package	190
Policy Info screen for commercial package	190
Line Selection screen for commercial package	192
Locations screen for commercial package	192
Line of business screens for commercial package	192
Modifiers screen for commercial package	193
Risk Analysis screen for commercial package	193
Quote screen for commercial package	193
Forms screen for commercial package	194
Payment screen for commercial package	194
Commercial package object model	195
Commercial package product model	195
24 Commercial property	197
Working with commercial property	197
Add locations and buildings in commercial property	197
Copying coverages to other buildings in commercial property	198
Add blanket coverages in commercial property	198
Configure blankets for commercial property in the product model	199
Commercial property screens	199
Policy Info screen for commercial property	199
Buildings and Locations screen for commercial property	201
Blankets screen for commercial property	202

Modifiers screen for commercial property	202
Risk Analysis screen for commercial property	202
Policy Review screen for commercial property	203
Quote screen for commercial property	203
Forms screen for commercial property	204
Payment screen for commercial property	204
Commercial property object model	204
Line entity in commercial property	205
Location entity in commercial property	206
Building entity in commercial property	206
Coverage entities in commercial property	206
Modifier entity in commercial property	207
Blanket entities in commercial property	207
Configuring copying coverages for commercial property	208
25 General liability.....	209
General liability overview	209
General liability policy basis examples	209
General liability screens	211
Qualification screen for general liability	211
Policy Info screen for general liability	211
Locations screen for general liability	213
Coverages screen for general liability	213
Exposures screen for general liability	214
Modifiers screen for general liability	215
Risk Analysis screen for general liability	215
Policy Review screen for general liability	216
Quote screen for general liability	216
Forms screen for general liability	217
Payment screen for general liability	217
General liability object model	217
Line entity in general liability	218
Coverage entity in general liability	218
Modifier entity in general liability	218
26 Homeowners	219
Homeowners overview	219
Homeowners dwellings per policy	220
Homeowners jurisdictions	220
Schedule endorsements for homeowners	220
Forms scope for homeowners	220
Coverages, exclusions, and conditions for homeowners	220
Authority profiles and underwriting rules for homeowners	220
Full application quote for homeowners	221
Homeowners rating	221
Homeowners offerings	221
Homeowners integrations	221
Homeowners wizard flow	221
Homeowners screens	222
Qualification screen for homeowners	222
Policy Info screen for homeowners	223
Dwelling screen for homeowners	225
Dwelling Construction screen for homeowners	225
Coverages screen for homeowners	226
Modifiers screen for homeowners	226
Risk Analysis screen for homeowners	226

Policy Review screen for homeowners	227
Quote screen for homeowners	227
Forms screen for homeowners	228
Payment screen for homeowners	228
Field validation in homeowners	228
Quote process for homeowners	229
Homeowners rating premium	229
27 Homeowners configuration	231
Homeowners data model	231
Homeowners coverables and coverages data model diagram	231
Homeowners cost data model diagram	232
Additional homeowners coverables data model diagram	235
Entities related to coverables in homeowners	238
Entities related to coverages in homeowners	239
Entities related to costs in homeowners	240
Homeowners typelists	240
Policy type and coverage form columns in homeowners	241
Configuring the homeowners product model	241
Homeowners coverage categories	242
Homeowners validation classes	243
Configuring underwriting issues for homeowners	244
Configuring forms for homeowners	244
Homeowners integrations and plugins	244
Configuring homeowners rating	244
Protection class code for homeowners	245
Homeowners insurance to value	245
28 Inland marine	247
Inland marine overview	247
Inland marine screens	248
Policy Info screen for inland marine	248
Coverage part selection screen for inland marine	250
Buildings and Locations screen for inland marine	250
Accounts Receivable screen for inland marine	250
Contractors Equipment screen for inland marine	251
Signs screen for inland marine	251
Risk Analysis screen for inland marine	251
Policy Review screen for inland marine	252
Quote screen for inland marine	252
Forms screen for inland marine	253
Payment screen for inland marine	253
Inland marine object model	253
Inland marine product model	254
29 Personal auto	257
Personal auto overview	257
Motor vehicle records in personal auto	257
Working with personal auto	259
Copy coverages to other vehicles	259
Personal auto screens	259
Policy Info screen for personal auto	259
Drivers screen for personal auto	261
Vehicles screen for personal auto	262
PA Coverages screen for personal auto	263
Risk Analysis screen for personal auto	263

Policy Review screen for personal auto	264
Quote screen for personal auto	264
Forms screen for personal auto	264
Payment screen for personal auto	265
Personal auto object model	265
Personal auto object model overview	265
PersonalVehicle entity in personal auto	266
VehicleDriver entity in personal auto	267
PolicyDriver entity in personal auto	267
Driver entity in personal auto	267
Coverages in personal auto	267
Modifiers in personal auto	268
Motor vehicle record object model in personal auto	268
Configuring personal auto	271
Configuring copy coverages for personal auto	271
Configuring personal auto motor vehicle records	271
30 Workers' compensation	273
Key features of workers' compensation	273
Policy term in workers' compensation	273
Jurisdictions in workers' compensation	274
Workers' compensation options	276
Workers' compensation screens	277
Qualification screen for workers' compensation	278
Policy Info screen for workers' compensation	278
Locations screen for workers' compensation	280
WC Coverages screen for workers' compensation	280
Supplemental screen for workers' compensation	283
WC Options screen for workers' compensation	283
Risk Analysis screen for workers' compensation	284
Policy Review screen for workers' compensation	285
Quote screen for workers' compensation	285
Forms screen for workers' compensation	286
Payment screen for workers' compensation	286
Workers' compensation object models	286
Workers' compensation object model overview	286
Coverage entities in workers' compensation	288
Modifier entity in workers' compensation	288
Employee entities in workers' compensation	288
Jurisdiction entity in worker's compensation	289
Retrospective rating plan entity in workers' compensation	289
Part 5	
Additional features of PolicyCenter	
31 Policy file	293
Policy file overview	293
Overview of copying data between policies	293
Overview of split and spin-off policies	294
Earned premium	295
Loss ratio	295
Policy file screens, menus, and actions	295
Info bar in policy file	296
Actions menu in policy file	296
Policy contract in policy file	296

Tools menu in policy file	296
Policy Summary in policy file	297
Working with policies	298
Copy data from one policy to another	298
Split an existing policy into two policies	299
Configuring policies	300
Core entities associated with policies	300
Policy object model overview	300
Policy term and policy period entities	302
Configuring copy data for a line of business	303
Configuring split and spin-off policies	304
Split and spin-off policies object model	304
Split and spin-off methods in Gosu classes	304
32 Account file	305
Account overview	305
Account security	305
Related accounts	305
Moving and rewriting policies between accounts	306
Merging accounts	306
Service tier in accounts	307
Account status	307
Account screens	307
Account Summary	309
Account Holder Summary	311
Billing screen for accounts	312
Account actions	313
Working with accounts	314
Search for an account	314
Account search minimum search criteria	314
Create an account	314
Track your accounts	315
Move a policy from one account to another	315
Rewrite policies from one account to an existing account	316
Merge accounts	317
Add an account relationship	318
Modify an account relationship	318
Remove an account relationship	319
Search for accounts with a shared contact	319
Configuring accounts	319
Account object model	319
Account rule sets	321
Account web service	321
Withdraw accounts	321
Configuring moving policies between accounts	321
Moving policies between accounts data model	321
Moving policies in the Account plugin	322
Configuring account relationships	322
Account relationship object model	322
Account relationship typelist	323
Account relationship methods in the Account plugin	323
Account relationship rule sets	323
Account relationship methods in Gosu classes	323
33 Locations	327
Location overview	327

Types of location information	327
Geocoding locations	327
Synchronizing and revisioning location information	328
Changing location information	328
Hiding location information on accounts	329
Location object model	329
Account synchronization classes for locations	330
Working with account locations	331
Add a new location	331
Edit a location	332
Make a location primary	332
Make a location active or inactive	332
Working with policy locations	332
Viewing locations on bound policies	333
Editing location information on a policy transaction	333
Adding or removing locations at the policy level	333
Location numbering	333
34 Activities	335
Activities overview	335
Activities creation and assignment	336
Activity ownership	336
Activity escalation	336
Localizing activity patterns	336
Working with activities	336
Create an activity	337
Assign an activity	337
Complete or skip an activity	337
Select an activity from a queue	338
Activity patterns	338
Activity pattern components	338
Creating and editing activity patterns	339
Using Gosu to edit activity patterns	340
Activity object model	340
Activity batch process	341
35 Notes	343
Differences between notes and documents	343
Working with notes	344
Searching for notes	344
Viewing notes	344
View notes related to an activity	344
Edit a note	345
Delete a note	345
How to print a note	345
Create a note	345
Create a note from a note template	345
Creating a note in an activity	346
Note security	346
Permissions related to notes	346
Configuring notes and note templates	346
Note plugin interfaces	346
Note fields	347
Creating a note template	347

36 Contingencies	349
Contingency overview	349
Working with contingencies	349
Viewing contingencies	349
Creating contingencies	350
Adding notes, documents, and activities to contingencies	351
Changing contingency status	351
37 Contacts	353
Contact overview	353
Centralized view of contacts on the Contact tab	353
Sharing contacts with a contact management system	354
Contacts and roles	354
Types of contact information	354
Revisioning contact information in policies	355
Linking an address between multiple contacts	357
Working with the Contact tab	357
Create a new contact	358
Search for a contact	358
Contact search results	358
Selecting recently viewed contacts	359
View contact file details	359
Create new account from contact	359
Viewing accounts associated with a contact	360
View policies associated with a contact	360
View policy transactions associated with contact	360
View claims associated with a contact	361
Contact File Claims screen	362
Viewing billing information for a contact	362
Contact tab behavior	363
Working with contacts in policies and accounts	363
Adding a contact to an account	363
Adding a contact from the address book	363
Edit a contact	364
Remove a contact from an account	365
Change the active status on a contact from an account	365
Adding or removing a contact from a policy	365
Add or remove a contact role	365
Working with linked addresses	366
Changing revised contact information in future-dated policy change	367
Changing revised contact information in a back-dated policy change	369
Viewing revised contact information in the account	371
Contact object model	371
Linked addresses object model	372
Contact roles for accounts and policies	373
Account synchronization classes for contacts	374
Configuring contacts	376
Gosu classes for contacts	377
Plugins for contacts	377
Contact batch process	378
Configuring the Contact tab	378
Gosu classes that implement features for the Contact tab	378
PCF files in the Contact tab	378
Configuring contact roles	379
Data model patterns for contact roles	380

Creating entities that define the new subtypes	380
Create an implementation of the contact configuration plugin	382
Add display key and entity name	382
Modify PCF files and Gosu classes	384
Adding a revised field to a contact	384
Defining the revised field on the account audit contact	384
Defining the revised field on the policy audit contact	385
Defining field as syncable on the policy contact role	386
Define the field as syncable on the account contact role	388
Extend entity and Gosu class for future-date policy changes	389
Add get and set methods to the policy contact role	390
Add revised field to PolicyCenter user interface	391
Configuring linked addresses for contacts	391
38 Quoting and rating	393
Working with quotes	393
Working with the Quote screen	393
Entities associated with costs and transactions	393
Cost delegate	394
Transaction delegate	397
Policy period fields for costs and transactions	398
Cost and transaction model for businessowners line	398
Cost and transaction model for commercial auto line	400
Cost and transaction model for commercial property line	400
Cost and transaction model for general liability line	401
Cost and transaction model for inland marine line	403
Cost and transaction model for personal auto line	404
Cost and transaction model for workers' compensation line	406
Calculating transactions	406
Internal tools for rating: financial transactions screen	408
39 Rating overrides	411
Rating overrides permissions	411
Underwriting issues for rating overrides	412
Processing rating overrides across policy transactions	412
Rating overrides in the user interface	412
Rating overrides on the Quote screen	412
Rating Overrides screen	413
Override a rating	413
Adding rating overrides to a line of business	414
Enable Override Rating button	414
Create panel set for rating overrides	414
Update the rating engine to handle overrides	415
40 Quote purging	417
Quote purging overview	417
Quote purging configuration business cases	418
Quote purging: what gets purged or pruned?	418
41 Quote cloning for business intelligence	421
Quote cloning overview	421
Quote cloning business example	422
42 Improving quoting and rating performance	423
Improving performance with parallel rating	424
Improving performance with parallel product model synchronization	425

Improving performance with asynchronous quoting	425
Asynchronous quoting flow	425
Asynchronous quoting threshold	426
Asynchronous quoting for multi-version and side-by-side?	426
Asynchronous quoting user interface	426
Improving performance with two-step quoting	428
One and two-step quoting workflows	428
43 Risk assessment with Spotlight	431
Risk assessment with Spotlight overview	431
Spotlight Integration Requires HTTPS	431
Risk assessment user interface	431
Using the risk assessment Location Information screen	432
Update risk evaluations	432
44 Policy forms	435
Policy forms overview	435
45 Policy data spreadsheet import/export	437
Large policy workflow using policy data spreadsheet import/export	437
Policy data spreadsheet import/export in commercial property	438
Using spreadsheets generated by policy data spreadsheet import/export	439
Export buildings and locations to a spreadsheet	439
Using exported spreadsheets	440
Import spreadsheet into policy transaction	443
46 Product model overview	445
Product model representation	445
Products overview	445
Policy line pattern overview	446
Coverage pattern overview	446
Product model categories	446
Cov coverages, exclusions, conditions, and coverables overview	447
Product model existence	447
Coverage term pattern overview	448
Product model availability overview	448
Determining the reference date for availability	449
Making a product model pattern available by policy transaction type	449
Grandfathering in the product model	449
Reloading availability in PolicyCenter	450
Offerings in the product model	450
Filtering the product model in availability	450
Offering question sets	451
Select and change an offering	451
Schedules overview	451
47 Policy revisioning	453
What is a policy revision?	453
Basic revisioning structure of a policy	455
The policy period and effective date fields	457
Structure of revisioning across effective time	459
Unbound policy revisions	462
Slice mode and window mode overview	462
Slice mode APIs	463
Window mode API overview	464
Safely accessing foreign keys with slice mode	466

Version list API	467
Working with window mode (unsliced) objects	471
Comparing window mode edits to slice edits	472
Out-of-sequence jobs	472
Out-of-sequence job user interface	473
Back-dated versus out-of-sequence job	474
Validation issues and out-of-sequence jobs	474
Preempted jobs	474
Create a preempted job	476
Applying changes to future renewals	477
Revisioning rewrite jobs	478
Summary of revisioning terminology	479
Revisioning properties reference	480
Revisioning properties on a policy	480
Revisioning properties on a policy period	480
Revisioning properties on PolicyPeriod subobjects	481
Details of merging and applying changes	483
Applying changes details	483
Merging changes details	485
Policy differences between revisions	486
48 Multicurrency features	487
Multicurrency overview	487
Single currency and multicurrency in PolicyCenter	487
Multicurrency terminology	487
Types of multicurrency policies	488
Exchange rate for multicurrency policies	488
Multicurrency and rating	489
Multicurrency and reinsurance	491
Multicurrency and basis units	492
Multicurrency object model	492
Multicurrency properties	492
Multicurrency in a policy line	493
Multicurrency user interface	494
Multicurrency fields on policy transactions screens	494
Multicurrency fields on the Contact screen	496
Multicurrency fields on the Account screen	496
Multicurrency fields on the Organization screen	496
Multicurrency fields on the Producer Code screen	496
Multicurrency field in reinsurance	497
Multicurrency fields for underwriting authority	497
49 Archiving in PolicyCenter	499
Archiving overview	499
Key features of archiving	499
Advantages of archiving	500
Impact archiving on your PolicyCenter configuration	500
More information on archiving	500
Archiving policy terms	500
Policy terms that PolicyCenter does not archive	501
Entities retained after archiving	501
Run Archive Policy Term batch process	502
Exclude policy from archiving	503
Searching for archived policy periods	503
Search archived policy periods	503
Desktop and Team tabs and archiving	504

Retrieving archived policies	504
Request retrieval of an archived policy	504
Retrieve archived policies that have been requested	505
50 Personal data destruction	507
Encapsulation of business logic for retention and destruction	507
Notification of data protection officer on errors or conflicts	508
Wide-swath data destruction	508
Individual-entity data destruction	508
Integration with other systems	508
Notification of downstream systems	509

Part 6

Rating Management

51 Rating Management concepts	513
Rating Management overview	513
Key features of Rating Management	514
Rating Management by line of business	515
Sample data for Rating Management	515
Rate flow design	515
Rate tables in Rating Management	516
Rate table overview	516
Rate table definition	517
Rate table normalization for overlapping ranges	519
Considerations for rate table size	520
Matching a factor in the rate table	521
Rate routines in Rating Management	526
Rate routine design	527
Rate routines that do not calculate properties on the cost	527
Rate routine versions	528
Rate routine variant identifiers	528
Rate routine steps	528
Considerations for rate routine functions	529
Accessing entity properties and class extensions	530
Parameter sets in Rating Management	530
Parameter set design	530
Combine similar parameter sets with wrappers	531
Rate books in Rating Management	531
Managing rate books and rate tables	531
Changes to parameters in rate table definitions	533
Changing factors in rate table definitions	534
Other types of changes in rate table definitions	535
Versioning in Rating Management	535
Selecting the rate book edition during policy rating	535
Rate book matching process	536
Filtering for the most appropriate rate book	536
Filtering for cascaded lookup rate books	537
Example of cascaded lookup in rate book	538
Overlapping effective policy periods and rate-as-of date in rate book	539
Rate book lifecycle and moving to production	540
Synchronize development rate books with production	541
Development environment with Rating Management	541
Stage environment with Rating Management	542
Production environment with Rating Management	542

Rating worksheets in Rating Management	542
Extracting and purging rating worksheets	543
Testing Rating Management	543
Impact testing for Rating Management	543
General guidelines for testability of Rating Management	544
Reducing Rating Management components	545
Rating Management component applies to all	545
Combine similar rate routines and parameter sets	545
52 Rating Management user interface	547
Preparing to use Rating Management	547
Working with rate books	547
Search for rate book	547
Add new rate book	548
Delete rate book	548
Rate Book screen	549
Rate book status and available actions	551
Rate book actions and permissions	551
Merge rate books	552
Export rate book to spreadsheet	553
Importing and exporting rate books to XML	553
Importing rate tables from spreadsheet	556
Working with rate table definitions	557
Search for rate table definition	557
Add a new rate table definition	558
Rate Table Definition screen	558
Working with the Rate Table editor	563
Selecting a rate table	563
Rate Table screen	563
Edit rate table content in PolicyCenter	564
Edit rate table content in Excel	564
Rate table update validations	565
Excel rate table import validations	565
Working with rate routines	566
Access rate routines	566
Add new rate routine	566
Delete rate routine	567
Actions on rate routines	568
Adding steps to a rate routine	568
Instruction and operand types in rate routine steps	571
Specify function as operand in rate routine step	577
Create rate routine for another jurisdiction	577
In rate routine, specify coverage as flat-rated	578
Editing long rate routines	578
View rating worksheets	579
Working with parameter sets in Rating Management	579
Access parameter sets in Rating Management	580
Add parameter set in Rating Management	580
Alter parameters in parameter set in Rating Management	580
Adding policy line rate modifiers to a parameter set	581
Working with impact testing in Rating Management	581
Impact testing warnings and recommendations	581
Permissions for impact testing	582
Test periods generated by impact testing	582
Generate test periods in impact testing	582

Examples of working with Rating Management	585
Creating and using a rate table with a multiple factors.	585
Adding parameters to an in-use rate table definition	587

Part 7**Reinsurance Management**

53 Reinsurance Management concepts	593
Reinsurance program basics	593
Using reinsurance programs to serve business goals	594
Reinsurance program design	594
Reinsurance program example	596
Reinsurance agreements	598
Treaties	598
Facultative agreements	599
Proportional agreements	599
Non-proportional agreements	602
Summary of agreement types	606
How Reinsurance Management links reinsurance to policies.	607
How Reinsurance Management attaches programs to policies	607
How Reinsurance Management attaches agreements to policies	607
Calculating ceded premiums in Reinsurance Management	610
Calculating ceded premiums in Reinsurance Management	611
Example: ceded premiums in Reinsurance Management	612
Gross retention	613
Shared reinsurance agreements	614
Differential rates of commission in reinsurance	614
Location groups and reinsurance	615
54 Reinsurance Management user interface	617
Working with the Reinsurance tab	617
Search for agreements in reinsurance	617
Search for all agreements in reinsurance	618
Create a new treaty in reinsurance	618
Create a new program in reinsurance	618
Edit a program in reinsurance.	618
Disable a program that has attached policies in reinsurance	619
Create a new facultative agreement in reinsurance	619
Validate an agreement in reinsurance	619
Make an agreement active in reinsurance.	620
Working with Reinsurance Management in policies	620
Add reinsurance to a policy	620
Create a location group	621
View ceded premiums.	622
Modify the gross retention	623
Adding or linking to a facultative agreement	623
Edit ceding parameters	624
Reinsurance Management screens	625
Treaty or Facultative Agreement screen.	625
Reinsurance Program screen	630
Search Agreements screen for reinsurance	632
Search Programs screen for reinsurance	632
Reinsurance screen in the policy file	633

Part 8

Underwriting authority

55 Underwriting authority concepts	639
Underwriting authority overview	639
Underwriting authority components	640
Choosing between validation and underwriting authority	640
Underwriting issue flow	640
Implement underwriting authority	641
56 Underwriting rules	643
Overview of underwriting rules	643
Business rule execution flow	644
Accessing business rules	644
Setting up business rules	644
Business rule states	644
Guidelines for designing underwriting rules	645
Underwriting issues with values	647
Checking sets and blocking points for underwriting issues	649
Underwriting referral reasons raise underwriting issues	652
Approvals of underwriting issues	653
Underwriting Rules screen	655
Externally managed rules	655
Clone business rules	655
Promote business rules	655
View business rule history	656
Enabling or disabling a business rule	656
Deleting a business rule version	657
Underwriting Rule screen	657
Rule Details tab on Underwriting Rule screen	657
Advanced Info tab on Underwriting Rule screen	659
Entering expressions in business rules	660
Working with business rule variables	663
Specifying business rule conditions	665
Operations in business rule conditions	665
Specifying underwriting issue details	667
Managing business rule export and import	668
Exporting and importing data lookup tables	668
Exporting and importing underwriting issue types	668
Manage Data Lookup Tables screen	668
Create New Lookup screen	669
57 Underwriting issues	671
Underwriting issues overview	671
Working with underwriting issues	671
Underwriting issues on the Risk Analysis screen	671
Risk Approval Details popup for risk analysis	672
Add underwriting referral reasons	674
58 Document management	677
Document storage overview	677
Document metadata properties	678

Part 9

Documents

58 Document management	677
Document storage overview	677
Document metadata properties	678

Working with documents	679
Viewing documents	679
Searching for documents	680
Create a new document	681
Edit content for a document	683
Edit metadata properties of a document	684
Hiding a document	684
Delete a document	684
Configuring and integrating document management	685
Document security	685
Configuration parameters for document management	685
Document management integration	686
Creating a document template	687
59 Smart Communications for PolicyCenter	689
Third-party software requirements for Smart Communications for PolicyCenter	690
Special features in Smart Communications for PolicyCenter	690
Document types in Smart Communications for PolicyCenter	690
Network flows in Smart Communications for PolicyCenter	691
Synchronous draft document creation flow in Smart Communications for PolicyCenter	691
Document production flow with Smart Communications for PolicyCenter	694
Asynchronous bulk document production in Smart Communications for PolicyCenter	695
Attaching documents to objects in Smart Communications for PolicyCenter	695
Specifying when Smart Communications for PolicyCenter creates documents	695

Part 10

PolicyCenter administration

60 Security: roles, permissions, and the community model	699
Community model overview	699
Producers in the community model	700
Types of security	701
Role-based security	701
Data-based security for accounts and policies	702
Data-based security for the community model	705
System and application permissions	706
Security restrictions using the status field	706
Producer of record and producer of service	707
Producers of service can edit the account	707
Changing the producer	707
Adding a third producer	707
Managing the PolicyCenter community	708
Understanding internal and external administration	708
Security object models	708
Object model for producer codes	709
Working with users and security	709
View permissions on selected roles	709
Create a permission	710
Remove a permission	710
Add a permission to a role	710
Add a new role	710
Remove a role	711
Turn on producer code security	711
Working with regions	711
Designating a client data integration handler	711

Working with affinity groups	712
Security and configuration scenarios related to producer codes	713
Producer codes assigned by level	713
Producer codes assigned by line of business	713
Producer codes assigned by level and line of business	714
Producer codes roles customized by user	714
Producer code security for agents working for multiple agencies	715
Producer code currency	716
Security Dictionary	717
Access control for documents and notes	717
Working with access control for documents and notes	718
su, the Super User	719
61 Authority profiles	721
Authority profile overview	721
Working with authority profiles	722
View or edit an authority profile	723
Assign an authority profile to a user	723
62 Team management	725
Team management overview	725
Groups and the Team tab	725
Team tab user categories	726
Reporting categories on the Team tab	726
Working with the Team tab	727
Assign activities on the Team tab	727
Assign submissions, renewals, and other policy transactions	727
63 Policy holds administration	729
Policy holds overview	729
Underwriting holds for natural disasters	729
Policy holds for regulatory changes	730
Specifying policy holds	730
Prevent back-dating policy transaction to avoid underwriting hold	731
Working with policy holds	731
Example: creating a simple policy hold	731
Work with policy hold actions	734
Deleting or disabling a policy hold	735
Policy Holds screen	735
Create or edit a policy hold	736
Policy Hold Details tab	736
Policy Hold Regions tab	737
Copying a policy hold	737
Policy hold actions in renewals	737
Policy hold object model	738
Configuring policy holds	738
Policy hold permissions	739
Policy hold authority grant	739
Policy hold underwriting issue types	739
Policy hold Gosu classes	739
Policy Hold batch process	740
Importing and exporting policy holds	740
64 Holidays and business weeks	741
Specifying holiday dates	741
Holiday types	741

Working with holidays, weekends, and business weeks	742
Add a holiday.	742
Edit a holiday.	742
Delete a holiday	742
Create a new zone or type	743
Using Gosu methods to work with holidays	743
Gosu holiday methods that use zones and types	743
Business weeks and business hours	743
Business hours	744
Gosu methods for business hours	744
Holiday permissions	744
65 Policy form pattern administration.	745
About forms	745
Form basics	746
Working with form patterns	746
Search for form pattern	747
Add form pattern	747
Specifying removal or replacement forms for policy changes	748
Importing and exporting policy form patterns.	748
Form Pattern or New Policy Form screen	748
Basics tab for form patterns	748
Products tab for form patterns	749
Transaction Types tab for form patterns	750
Jurisdictions tab for form patterns.	750
Policy Change tab for form patterns	751
Inference tab for form patterns	753
Form configuration	754
Adding a custom inference class for form patterns	754
Configuring custom form inference	756
Configuring form inference by coverage part and policy type	757
Configuring generic form inference	757
Form pattern validation	758
66 Administration utilities	759
Importing and exporting data	759
Administering script parameters	759
Administering data changes	759
Importing and exporting policy data spreadsheets	760
Spreadsheet export formats user interface	760
Define export formats	760
Part 11	
External system integration	
67 Billing system integration.	765
Billing system integration overview	765
Organizations and producer codes in PolicyCenter and billing system.	766
Producer organizations in PolicyCenter and BillingCenter.	766
Multicurrency and producer organizations	766
Multicurrency and producer codes	767
Accounts in PolicyCenter and billing system	767
Multicurrency and accounts	768
Policies in PolicyCenter and billing system	768
Sending charges and other information to billing system	769

Payment screen in PolicyCenter	769
Contacts in PolicyCenter and billing system	769
Billing system and policy transactions that create a new policy period	769
Billing system and policy transactions that create midterm changes	771
Cancellations in the billing system integration	772
Reinstatements in the billing system integration	773
Renewals or rewrites in the billing system integration	773
Final audits in the billing system integration	774
Premium reporting in the billing system integration	775
Working with the billing system integration	775
Working with the Payment screen	775
View policy period in BillingCenter	779
Working with accounts from the billing system	779
Working with policies in the billing system integration	780
Multicurrency integration between BillingCenter and PolicyCenter	780
Limitations of multicurrency billing and policy integration	781
Multicurrency accounts in PolicyCenter and BillingCenter	781
Using commission plans to select currencies for producer codes	781
Single currency producer codes in PolicyCenter and BillingCenter	782
Multicurrency producer codes and producer organizations	782
Default billing and policy multicurrency integration	782
68 Claim system integration	783
Claim system integration overview	783
Accessing summary loss information from the claim system	784
Viewing loss claims for policies	784
Viewing claim on policy in claim system	785
Viewing loss claims from an account	785
Loss claims notification at renewal	786
Large loss notification from ClaimCenter	787
Permissions for working with claims	787
Restricted fields in claims	787
Claim Search plugin	787
69 Contact management system integration	789
Contact management system integration overview	789
Searching for contacts within a contact management system	790
New and updated contacts and contact management system	790
Pushing new and updated contacts to contact management system	791
Pushing new and updated contact to ContactManager	791
Adding a contact from the contact management system	791
New contact when integrated with contact management system	792
Detecting duplicates in the contact management system	793
Detecting duplicates when integrated with ContactManager	793
Duplicate contacts in PolicyCenter	796
Deleting, removing, and inactivating a contact	797
Customizing the contact management system integration	797
Contact management entry points into PolicyCenter	797
Integrating with multiple contact management systems	798

Guidewire InsurancePlatform™

Guidewire InsurancePlatform™ is the property and casualty industry platform that unifies software, services, and a partner ecosystem to power our customers' businesses. InsurancePlatform provides the standard upon which insurers can optimize their operations, increase engagement, drive smart decisions, innovate quickly, and simplify IT.

This documentation provides information on installation, implementation, administration, and customization for Guidewire InsurancePlatform™ software and SaaS offerings.

About PolicyCenter documentation

The following table lists the documents in PolicyCenter documentation:

Document	Purpose
<i>InsuranceSuite Guide</i>	If you are new to Guidewire InsuranceSuite applications, read the <i>InsuranceSuite Guide</i> for information on the architecture of Guidewire InsuranceSuite and application integrations. The intended readers are everyone who works with Guidewire applications.
<i>Application Guide</i>	If you are new to PolicyCenter or want to understand a feature, read the <i>Application Guide</i> . This guide describes features from a business perspective and provides links to other books as needed. The intended readers are everyone who works with PolicyCenter.
<i>Upgrade Guide</i>	Describes the overall PolicyCenter upgrade process, and describes how to upgrade your PolicyCenter configuration and database from a previous major version. The intended readers are system administrators and implementation engineers who must merge base application changes into existing PolicyCenter application extensions and integrations. The <i>Upgrade Guide</i> is published with the InsuranceSuite Upgrade Tools and is available from the Guidewire Community.
<i>New and Changed Guide</i>	Describes new features and changes from prior PolicyCenter versions. Intended readers are business users and system administrators who want an overview of new features and changes to features. Consult the “Release Notes Archive” part of this document for changes in prior maintenance releases.
<i>Installation Guide</i>	Describes how to install PolicyCenter. The intended readers are everyone who installs the application for development or for production.
<i>System Administration Guide</i>	Describes how to manage a PolicyCenter system. The intended readers are system administrators responsible for managing security, backups, logging, importing user data, or application monitoring.
<i>Configuration Guide</i>	The primary reference for configuring initial implementation, data model extensions, and user interface (PCF) files for PolicyCenter. The intended readers are all IT staff and configuration engineers.
<i>PCF Reference Guide</i>	Describes PolicyCenter PCF widgets and attributes. The intended readers are configuration engineers.
<i>Data Dictionary</i>	Describes the PolicyCenter data model, including configuration extensions. The dictionary can be generated at any time to reflect the current PolicyCenter configuration. The intended readers are configuration engineers.
<i>Security Dictionary</i>	Describes all security permissions, roles, and the relationships among them. The dictionary can be generated at any time to reflect the current PolicyCenter configuration. The intended readers are configuration engineers.
<i>Globalization Guide</i>	Describes how to configure PolicyCenter for a global environment. Covers globalization topics such as global regions, languages, date and number formats, names, currencies, addresses, and phone numbers. The intended readers are configuration engineers who localize PolicyCenter.
<i>Rules Guide</i>	Describes business rule methodology and the rule sets in Guidewire Studio for PolicyCenter. The intended readers are business analysts who define business processes, as well as programmers who write business rules in Gosu.
<i>Contact Management Guide</i>	Describes how to configure Guidewire InsuranceSuite applications to integrate with ContactManager and how to manage client and vendor contacts in a single system of record. The intended readers are PolicyCenter implementation engineers and ContactManager administrators.
<i>Best Practices Guide</i>	A reference of recommended design patterns for data model extensions, user interface, business rules, and Gosu programming. The intended readers are configuration engineers.

Document	Purpose
<i>Integration Guide</i>	Describes the integration architecture, concepts, and procedures for integrating PolicyCenter with external systems and extending application behavior with custom programming code. The intended readers are system architects and the integration programmers who write web services code or plugin code in Gosu or Java.
<i>Java API Reference</i>	Javadoc-style reference of PolicyCenter Java plugin interfaces, entity fields, and other utility classes. The intended readers are system architects and integration programmers.
<i>Gosu Reference Guide</i>	Describes the Gosu programming language. The intended readers are anyone who uses the Gosu language, including for rules and PCF configuration.
<i>Gosu API Reference</i>	Javadoc-style reference of PolicyCenter Gosu classes and properties. The reference can be generated at any time to reflect the current PolicyCenter configuration. The intended readers are configuration engineers, system architects, and integration programmers.
<i>Glossary</i>	Defines industry terminology and technical terms in Guidewire documentation. The intended readers are everyone who works with Guidewire applications.
<i>Product Model Guide</i>	Describes the PolicyCenter product model. The intended readers are business analysts and implementation engineers who use PolicyCenter or Product Designer. To customize the product model, see the <i>Product Designer Guide</i> .
<i>Product Designer Guide</i>	Describes how to use Product Designer to configure lines of business. The intended readers are business analysts and implementation engineers who customize the product model and design new lines of business.

Conventions in this document

Text style	Meaning	Examples
<i>italic</i>	Indicates a term that is being defined, added emphasis, and book titles. In monospace text, italics indicate a variable to be replaced.	A <i>destination</i> sends messages to an external system. Navigate to the PolicyCenter installation directory by running the following command: <code>cd installDir</code>
bold	Highlights important sections of code in examples.	<code>for (i=0, i<someArray.length(), i++) { newArray[i] = someArray[i].getName() }</code>
narrow bold	The name of a user interface element, such as a button name, a menu item name, or a tab name.	Click Submit .
monospace	Code examples, computer output, class and method names, URLs, parameter names, string literals, and other objects that might appear in programming code.	The <code>getName</code> method of the <code>IDoStuff</code> API returns the name of the object.
monospace italic	Variable placeholder text within code examples, command examples, file paths, and URLs.	Run the <code>startServer server_name</code> command. Navigate to http://server_name/index.html .

Support

For assistance, visit the Guidewire Community.

Guidewire customers

<https://community.guidewire.com>

Guidewire partners

<https://partner.guidewire.com>

Part 1

Introduction to PolicyCenter

PolicyCenter overview

PolicyCenter is a web-based underwriting and policy administration system designed for personal and commercial line insurers 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. Insurers 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, businessowners, and workers' compensation are all products.

While PolicyCenter comes with certain lines of business, it is flexible. You can customize the default lines of business to meet your business needs. You can also create your own lines of business in Guidewire Product Designer. (PolicyCenter includes Product Designer.)

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.

Product model

The PolicyCenter product model is at the core of its line of business configuration. It defines the products that insurers 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 Product Designer to create and manage your product model.

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. 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 Product Designer. The backing data model, code, and PCF pages required to support a line of business must be created by a developer using Guidewire 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
- Homeowners
- Inland marine
- Personal auto
- Workers' compensation

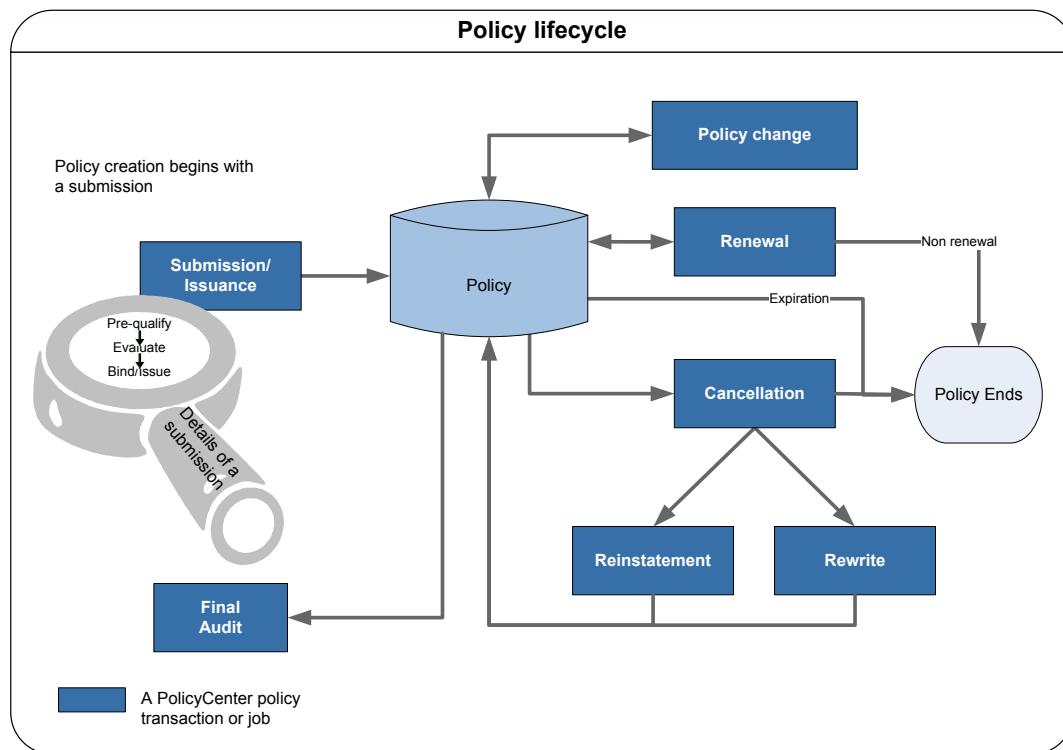
Note: The PolicyCenter default application is not a compliance system. Guidewire designed the product model so that you can maintain your lines of business within your jurisdictions, as necessary. For example, the system tables in the default application can accommodate multiple classifications per jurisdiction 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.

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.



Submission

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 change

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.

Renewal

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.

Cancellation and reinstatement

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.

Reinstatement

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.

Rewrite

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

Audit

The audit policy transaction lets the insurer 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 system** – 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 789.
- **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** – Insurers use a vendor management system to track vendor related information that an insurer uses. An example of a vendor might be an outside audit company.
- **Sales portal or application** – Some insurers 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 *suite* 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 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 765
- “Claim system integration” on page 783

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

Users	Typical Activities
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 2

PolicyCenter user interface

Navigating PolicyCenter

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

Logging into PolicyCenter

You log in to PolicyCenter by running the application and logging in with your user name and password.

PolicyCenter login requirements

Logging in to PolicyCenter requires the following:

- **A web browser** – For information about supported browsers, see 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 create a shortcut on your computer desktop that starts a web browser with that URL.
- **A user name and password**

Because PolicyCenter generates screens dynamically:

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

Log in to PolicyCenter

Procedure

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.

Result

If your login is successful, PolicyCenter displays your startup view, or landing page. In the default configuration, PolicyCenter initially 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 selecting **Preferences** from the Options menu . Your changes take effect the next time you log in.

In the **Preferences** worksheet you can specify:

- Email notification – 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.
- Password – Change your password.
- **Regional Formats** – Set the regional formats that PolicyCenter uses to enter and display dates, times, numbers, monetary amounts, and names.
- **Default Country** – Determines the settings for names and addresses.
- **Default Phone Region** – Determines how phone number entries are handled, especially the country code setting.
- **Startup Page** – Change the page that PolicyCenter displays when you log in by selecting a **Startup Page**.
- Recent activity – 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 from `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	<code>config.xml</code> parameter
Maximum Recent Accounts	<code>MaxRecentAccounts</code>
Maximum Recent Policies And Policy Transactions	<code>MaxRecentPoliciesAndJobs</code>
Maximum Recent Contacts	<code>MaxRecentContacts</code>

Changing interface settings

Change interface settings to control the behavior of certain functions in the user interface.

On the top tab bar, in the **Options**  menu, click **Settings**.

Appearance settings

Setting	Description
Disable outlines on focused elements	When not set, the input elements with focus have an extra outline to make them easier to identify.
Application font size	The base font size of the text used on the application screens.
Global spacing modifier	A multiplier that decreases (when less than 1) or increases (when greater than 1) the amount of whitespace surrounding visual elements.
Theme	The theme to use as the visual style of the application.
Left align top toolbars	Set to align the toolbar at the top of the screen to the left instead of the right.
Highlight changed values	Set to align the toolbar at the top of the screen to the left instead of the right.

Setting	Description
Force text shadows on	When set, dark text is displayed with a white shadow, and light text is displayed with a black shadow. This setting may assist with readability when there is low contrast between the text and its background.

Dates settings

Setting	Description
Use complex date picker options	Adds a Selected Day button to date pickers, which navigates the calendar to the currently selected date.
Use small date picker	Reduces the size of date pickers by reducing the font size and spacing.
Open date/time pickers on focus	Set to open a date picker automatically when you navigate to a date input. When this setting is not set, you must click the icon next to the date picker to open it.
Today button in date picker selects today and closes picker	Set to have the date picker close automatically after you click Today to set the date input to the current date.
Cap user input to max values for days, months, minutes, and hours	In a date input, automatically change values that are greater than the allowed values to the maximum allowed values.

General settings

Setting	Description
Always confirm browser navigation	When using browser navigation, such as by clicking the Back button, show a confirmation alert before changing the page.
Disable browser autocomplete	Disable the browser's autocomplete function to avoid having it suggest values for text inputs.
Scroll the screen to the top on any errors	When an input error occurs, automatically show the top of the screen, where the error message appears.
Navigating rows of a List Detail using up and down arrow keys also selects the row	When set, using the arrow keys to move up and down in a list detail view also selects the current row and shows its detail. When not set, using the arrow keys highlights a row but does not select it; to select it, click on it or press Enter .

Debug settings

Setting	Description
Highlight elements that are redrawn	Screen elements that are redrawn after an update pulsate for a short time. The visual effect identifies which elements were affected by the update.
Ignore PCF widths and heights	Render screen elements as if they do not have any width or height values set for them. Ignoring the width and height settings helps you see what they would look like without those values set.
Highlight widgets with PCF widths and heights	Surround all PCF widgets that have width and height attributes set with a highlight color. The highlighting helps you identify elements that have widths and heights explicitly set.
Show widget types as inline titles	Places a title near each PCF widget that shows its widget type. The titles help you identify the widgets on a page.

Currency settings

Setting	Description
Enable macro characters in currency inputs	Enhances the input and display options for currency values. See also “Using the currency macro in currency fields” on page 44.
Include the currency symbol when copying an amount	When selecting and copying the text in a currency element, specifies whether the currency symbol is included.
Show 0 as the currency input placeholder for null values	For currency elements, whether 0 is shown when the value is null. If not set, the element is empty.

Change the visual theme

You can choose a new theme for the application, which changes the visual appearance and behavior.

About this task

When you set a theme, the application immediately changes to reflect that theme.

Procedure

1. On the top tab bar, in the **Options**  menu, click **Settings**.
2. In the **Appearance** section, in the **Theme** drop-down list, click the theme to use.

Data entry support for input fields

As you type in data for some types of input fields, PolicyCenter formats the data appropriately for the field.

See also

- “Using the currency macro in currency fields” on page 44
- “As-you-type formatting support for input fields” on page 45
- “Highlight changed values” on page 46

Using the currency macro in currency fields

PolicyCenter includes a currency macro, which is a user interface feature that converts alphanumeric values entered in currency fields into numeric values. For example, in the base configuration, the user can enter an alphanumeric like **1.54k**, and the currency macro immediately converts it to **1,540**.

The characters that you enter depend on your regional format and whether the currency accepts decimal values. The macro can support up to three decimal values, but the number you can enter might be limited by the currency field.

In the base configuration, this macro supports the following values:

- k for thousand
 - If the regional format is United States (English), entering 1.54k produces 1,540.
 - If the regional format is Germany (German), entering 1,54k produces 1.540.
 - If the regional format is France (French), entering 1,54k produces 1 540.
- m for million
 - If the regional format is United States (English), entering 1.54m produces 1,540,000.
 - If the regional format is Germany (German), entering 1,54m produces 1.540.000.
 - If the regional format is France (French), entering 1,54m produces 1 540 000.
- b for billion
 - If the regional format is United States (English), entering 1.54b produces 1,540,000,000.
 - If the regional format is Germany (German), entering 1,54b produces 1.540.000.000.
 - If the regional format is France (French), entering 1,54b produces 1 540 000 000.
- t for trillion
 - If the regional format is United States (English), entering 1.54t produces 1,540,000,000,000.
 - If the regional format is Germany (German), entering 1,54t produces 1.540.000.000.000.
 - If the regional format is France (French), entering 1,54t produces 1 540 000 000 000.

Enabling and disabling the currency macro

In the base configuration, the currency macro is enabled by default.

To disable the currency macro, log in to PolicyCenter and click the Options menu , and in the **Settings** dialog under **Currency**, clear the check box for **Enable macro characters in currency inputs**.

To enable the currency macro, select the check box for **Enable macro characters in currency inputs**.

Configuring the currency macro keys

The macro characters are defined in the following display keys:

```
Web.Preferences.Currency.macroCharacter.Billion = b  
Web.Preferences.Currency.macroCharacter.Million = m  
Web.Preferences.Currency.macroCharacter.Thousand = k  
Web.Preferences.Currency.macroCharacter.Trillion = t
```

You can change these keys or localize them. If you do so:

- The new value must be a single character that can be typed with a single stroke on a keyboard.
- The macro produces groups of three for each order of magnitude. For example, if the locale is United States:
 - One thousand is formatted as 1,000.00.
 - One million is 1,000,000.00.
 - One billion is 1,000,000,000.00.
 - One trillion is 1,000,000,000,000.00.

See also

- “Options for setting regional formats” on page 47
- “Display keys and localization” in the *Globalization Guide*
- “Overview of display keys” in the *Configuration Guide*

As-you-type formatting support for input fields

As the user types in data for some types of input fields, the data is formatted appropriately for the field. This user input support is additional to currency macro support for currency fields.

In general, the formatting support does the following as the user types in the field:

Currency fields

Formats the user entry as currency, with appropriate group and radix characters for the locale.

Date fields

Formats the user entry as a date as set in the application.

Time fields

Formats the user entry as a time as set in the application.

Fields with an input mask

Formats by adding the input mask characters automatically as the user types.

See also

- “Using the currency macro in currency fields” on page 44
- “Field validator definitions” in the *Configuration Guide*
- “<DateFormat> and <TimeFormat> elements of a localization file” in the *Globalization Guide*

Highlight changed values

When you modify a data value, its background color changes so you can easily identify what data has been changed on the page.

About this task

If you revert the changed data back to its previous value, the highlighting is removed. The data remains highlighted until you click **Update** and submit the changes. If you do not want to highlight changed values, you can disable this behavior.

Procedure

1. On the top tab bar, in the **Options** ☰ menu, click **Settings**.
2. In the **Appearance** section, set or clear **Highlight changed values**.

Selecting language and regional formats in PolicyCenter

In Guidewire PolicyCenter, each user can set the following:

- The language that PolicyCenter uses to display 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 using the Options menu ☰ at the top, right-hand side of the PolicyCenter screen. On that menu, click **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 region or language.

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

PolicyCenter indicates the current selections for **Language** and **Regional Formats** by putting a check mark to the left of each selected option.

Options for setting the display language

In the base configuration, PolicyCenter is configured to use a single display language, English. To view another language, you must enable at least one additional language and configure PolicyCenter for that language. If your installation has more than one language, 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** menu and your user administrator has not set your language, PolicyCenter uses the language specified by your web browser. The configuration parameter **DefaultApplicationLanguage** specifies the primary language for PolicyCenter screens, but it does not specify the default browser language. In the base configuration, the primary language is U.S. English.

Options for setting regional formats

If your installation contains more than one configured region, you can select a regional format for that locale from the **Regional Formats** submenu. At the time you configure a region, 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 on the **Regional Formats** menu. The base configuration defines the following locale types:

- | | |
|--|---|
| <ul style="list-style-type: none">• Australia (English)• Canada (English)• Canada (French)• France (French) | <ul style="list-style-type: none">• Germany (German)• Great Britain (English)• Japan (Japanese)• United States (English) |
|--|---|

Unless you select a regional format from the **Regional Formats** menu, PolicyCenter uses the regional formats of the default region. The configuration parameter **DefaultApplicationLocale** specifies the default region. In the base configuration, the default region is **en_US**, United States (English). If you select your preference for region from the **Regional Formats** menu, you can later use the default region again only by selecting it from the **Regional Formats** menu.

See also

- *Globalization Guide*

My Summary

General use

The screen serves as your "to do" list. It displays your:

- Activities
- Submissions
- Renewals
- Change Requests

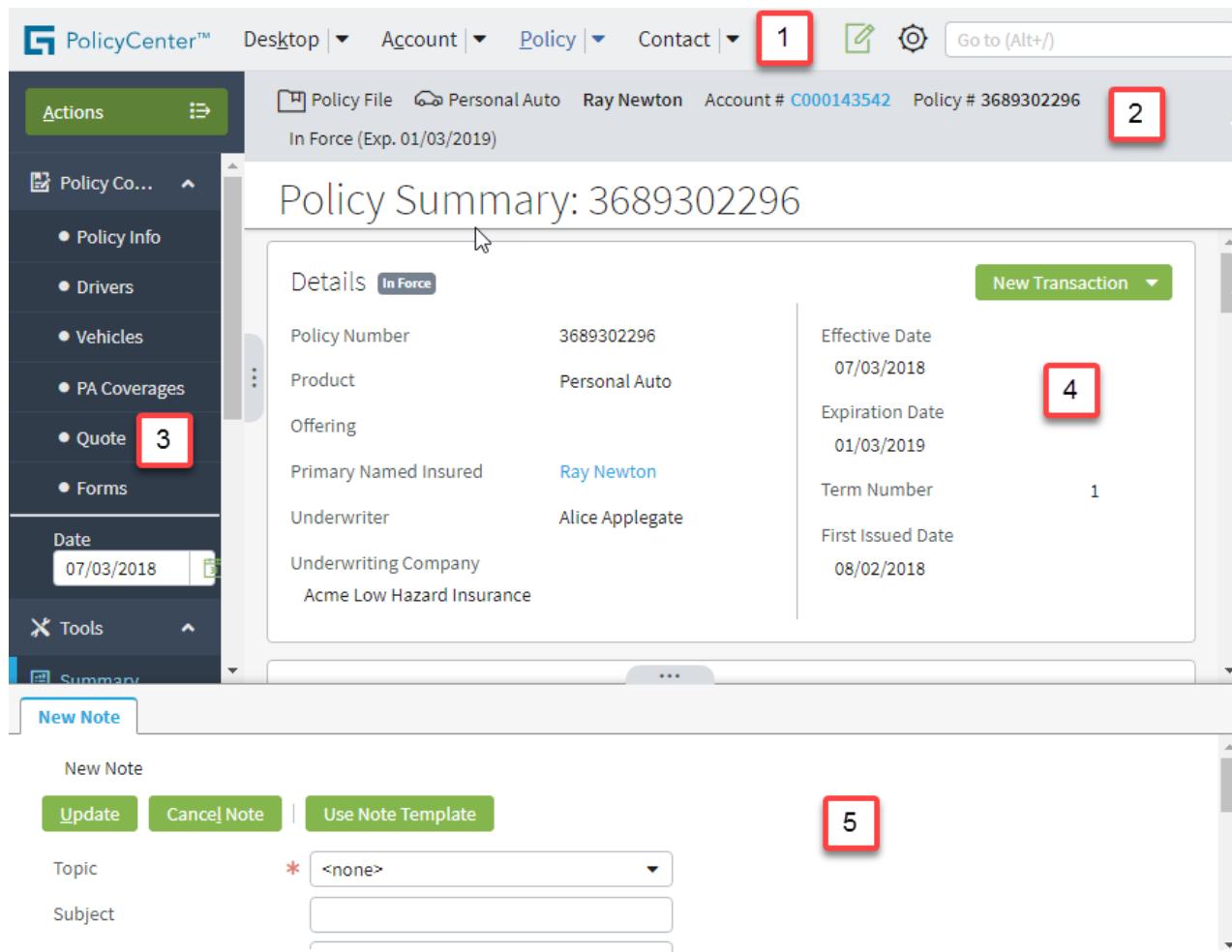
It is meant for underwriters, but it appears the same for all users, and serves as a hub for work - you can keep coming back to it throughout your day.

How to access

It appears when you log onto PolicyCenter, unless you have a different screen configured as your starting location. You can also access it by clicking **Desktop** in the top menu.

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

- | Area | Description |
|------|--|
| 1 | The Tab bar contains: <ul style="list-style-type: none"> Tabs QuickJump box is the text box that displays Go to (Alt+I). For more information, see “QuickJump” on page 61. Unsaved Work list. For more information, see “Saving your work” on page 79. 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. |
| 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. |

Area	Description
	The Date field displays the <i>as of</i> date of the policy term. PolicyCenter displays the policy data effective as of this date.
	In most cases, when viewing a bound policy, the Quote screen displays the premium of the bound policy, regardless of the date.
	If you enter this screen by selecting a policy transaction on the Account File Policy Transactions screen, PolicyCenter displays the policy transaction with the initial quote. If you change the date, PolicyCenter switches to a mode that displays the bound policy as of a certain date. In this mode, PolicyCenter displays the bound quote, whatever the 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.

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

My Activities screen

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 335

My Accounts screen

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 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** and **Producer Code - Basics** 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 305

My Submissions screen

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 726.

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.

The **My Submissions** screen has the following fields:

Field	Description
Primary Insured	The name of primary insured on the submission.
Effective Date	The <code>PolicyPeriod.EditEffectiveDate</code> from the policy period used to determine the Status. By default, this field is used to sort the list.
Quote Needed	For submission policy transactions, this field displays <code>Submission.DateQuoteNeeded</code> . For issuance policy transactions, this field is blank.
Transaction #	The transaction number.
Type	The type of submission.
Status	The status of the submission.
Issued	This field displays <code>Policy.Issued</code> .
Product	The product of the submission.
Producer	This field is visible if the user does not have the View Producer Desktop Details permission.
Underwriter	This field displays the user with the Underwriter role in the <code>Job.RoleAssignments</code> array.

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 base configuration, the **Producer** and **Producer Code - Basics** 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 89 and “Issuance policy transaction” on page 97
- *Configuration Guide*

My Renewals screen

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 screen” on page 50.

See also

- “Renewal policy transaction” on page 101

My Other Policy Transactions screen

The **My Other Policy Transactions** screen displays other policy transactions you created or are working on. This screen has the following fields:

Field	Description
Type	The type of the policy transaction.
Primary Insured	The name of the primary insured on the policy transaction.
Effective Date	The PolicyPeriod.EditEffectiveDate from the query used to determine the Status . By default, this field is used to sort the list.
Transaction #	The transaction number.
Status	The status of the policy transaction
Product	The product of the policy transaction.
Producer	This field is visible if user does not have the View Producer Desktop Details permission.

Field	Description
Underwriter	This field displays the user with the Underwriter role in the Job.RoleAssignments array.

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 screen” on page 50.

See also

- “Policy change transaction” on page 119
- “Cancellation policy transaction” on page 111
- “Reinstatement policy transactions” on page 127
- “Renewal policy transaction” on page 101
- “Premium audit policy transaction” on page 139

My Queues screen

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 “Select an activity from a queue” on page 338.

Configure search drop-down lists

About this task

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.

Use these steps to view or modify the search filters for the **My Activities** screen.

Procedure

- In Studio, open the `DesktopActivitiesLV` PCF file.
- Select the `activitiesFilter ToolbarFilter` PCF element.
- At the bottom of the screen, click the **Filter Options** tab.
- Select `gw.api.web.desktop/DesktopActivityFilters.filters()`.
The filter is defined by `gw.api.web.desktop/DesktopActivityFilters.filters()`.

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

See also

- *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 305.

For information about setting the number of recent accounts that PolicyCenter displays on the **Account** tab, see “Setting preferences” on page 42.

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 293.

For information about setting the number of recent policies and policy transactions that PolicyCenter displays on the **Policy** tab, see “Setting preferences” on page 42.

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 357
- “Configuring the Contact tab” on page 378
- “Setting preferences” on page 42 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

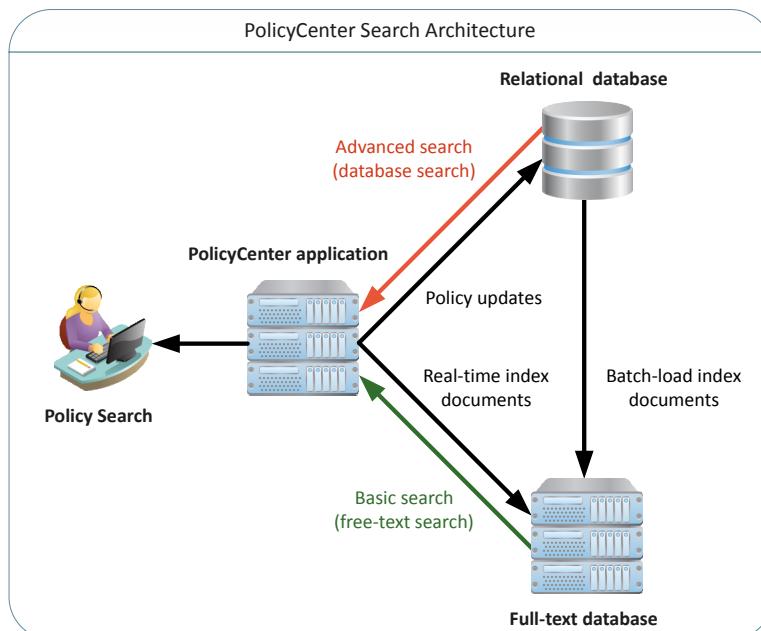
- “Reinsurance Management concepts” on page 593

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 67
- “Advanced search” on page 75

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 725.

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 insurer and see the insurer'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 697.

Changing the screen layout

You can adjust several aspects of the screen layout according to your own preferences.

Adjusting list views

You can change the default appearance of individual list views.

Change list view column order

About this task

You can change the order in which columns appear in a particular list view. This change remains in effect until you clear the layout preferences.

Procedure

1. Click and hold the left mouse button on the heading of the column that you want to move.
2. Drag the mouse pointer across the other column headings until it is between the two columns where you want to place the moved column.

If it is valid to move the column there, the column turns from gray to highlighted:



3. Release the mouse button.

See also

- “Clear layout preferences” on page 60

Sort list views

About this task

You can change the column on which a list view is sorted, and also the sorting direction. Note that you can sort on only one column at a time.

Procedure

Click the heading of a column to sort the list view on that column.

- To sort a list view on a particular column, click the column heading.
- To change the sort direction of a list view column, click the up or down arrow on the heading of the column on which the list is currently sorted:



Result

The up or down arrow is highlighted indicating the direction in which the list is sorted.

Hide and show columns in a list view

About this task

You can change which columns appear in a list view. This change remains in effect until you clear the layout preferences.

Procedure

1. Position the mouse pointer over the list view title row, and then click the drop-down menu icon  that appears at the right.
2. In the drop-down list, click the columns that you want to change:
 - To hide a column, clear the check box for the column.
 - To show a column, select the check box for the column.

See also

- “Clear layout preferences” on page 60

Grouping rows in a list view

When multiple rows in a list view have the same value for a particular column, you can arrange them to be listed together in a group. Each group appears in the list view under a new heading row for the common value. For example, if you have a list of activities, you can group them by their values in the **Due Date**. The related activities are then listed together under a heading for each due date.

The following figure shows rows grouped by **Due Date**.

Search Results				Print / Export	Assign	Skip			
<input type="checkbox"/>				Due Date		Priority		Status	
▲ Due Date:									
<input type="checkbox"/>						Normal		Open	
▲ Due Date: 08/08/2018									
<input type="checkbox"/>				08/08/2018		Normal		Open	
<input type="checkbox"/>				08/08/2018		Normal		Open	

You can group a list view only by one column at a time.

Arrange list view rows into groups

Procedure

1. Position the mouse pointer over the list view title, and then click the drop-down Columns icon  that appears at the right.
2. In the drop-down menu, click the **Group/Ungroup** icon .

Ungroup list views

Procedure

1. Position the mouse pointer over the list view title row, and then click the drop-down menu icon  that appears at the right.
 2. In the drop-down menu, deselect the **Groups/Ungroup** icon .
- To collapse or expand a group, click  to collapse or  to expand next to the group name.

Change the sidebar width

About this task

You can resize the sidebar to make it wider or narrower than it currently is. This change remains in effect until you clear the layout preferences.

Procedure

1. Position the mouse pointer over the right border of the sidebar. The pointer turns into a double arrowhead .
2. Drag the sidebar border to the new width.

See also

- “Clear layout preferences” on page 60

Managing layout preferences

PolicyCenter records many changes that you make to the screen layout and automatically saves them as layout preferences. For example, changes to the size, ordering, and visibility of list view columns are stored in the layout preferences. Once set, these preferences remain in effect, even if you log out and log back in.

Storage of layout preferences

Layout preferences are stored in your web browser’s local storage. Using the local storage has several implications:

- If you log in using a different browser, then PolicyCenter uses the preferences stored with that browser, if any, rather than your preferences set in the first browser.
- If another user logs in to PolicyCenter from your browser, their layout will reflect the preferences that are stored with your browser.
- Clearing the browser’s cookies also clears the saved layout preferences.
- Clearing the browser cache does not clear the saved layout preferences, and clearing the layout preferences has no effect on the browser’s cache.

Clear layout preferences

About this task

To reset your layout changes back to their defaults, you must clear the layout preferences. This action does not affect the web browser's cache.

Procedure

In the **Options**  menu, click **Clear Layout Preferences**.

QuickJump

QuickJump is a feature in the PolicyCenter user interface that can be used to perform navigation to a screen using the keyboard only. It is intended primarily for users who prefer to navigate without using a mouse.

QuickJump overview

The **QuickJump** box provides a fast way to navigate to a particular screen in the application.

Some of the PolicyCenter screens are:

- **Desktop** tab
- **Search** tab
- **Team** tab
- **Admin** tab

The **QuickJump** box can also retrieve and show information about a particular entity. In the base configuration, entities that PolicyCenter provides are **Policy** and **Account**. You can add additional entities.

Using QuickJump

The **QuickJump** box appears at the upper right corner of most PolicyCenter screens. The box is not available in pop-ups.



To use the box, position the cursor in it or use the shortcut key **Alt /**, and then enter a QuickJump command. To view a list of available commands, press the **Down Arrow** key.

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

The **QuickJump** box provides automatic command and parameter completion. Type the first few letters of a command, and the **QuickJump** box automatically provides a list of the possible commands. For example, type the letter **A** to list all commands or parameters that begin with the letter **A**.

Chaining QuickJump destinations together

It is possible to chain multiple QuickJump destinations together to jump to a specific screen.

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**, press the Spacebar, and then type 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.

QuickJump behavior in wizards

Wizards are typically used to advance sequentially through a series of steps. QuickJump can be used to skip steps and jump to a screen listed in the Sidebar.

QuickJump actions available in a wizard are active only when operating in the wizard. For example, it is not possible to jump from an account screen to a specific wizard screen.

When operating in a wizard, if you want to jump to another part of PolicyCenter, save your work before jumping. After the jump, if you did not save your work, your wizard work will be lost.

Configuring QuickJump

The **QuickJump** box can be configured in various way.

- You can add new commands that jump to newly-created screens.
- You can change existing **QuickJump** commands. For example, you can provide commands that users were accustomed to using on another system.
- You can remove the **QuickJump** box from the user interface.

You can use the XML Editor in Studio to configure the **QuickJump** box. In the **Project** window, navigate to **configuration→config→Page Configuration** and open **quickjump-config.xml** to edit QuickJump resources. Labels for a particular language are defined in the **display_LanguageCode.properties** file.

See also

- *Configuration Guide*
- *Globalization Guide*

QuickJump reference

The tables in the topics that follow list the QuickJump commands that PolicyCenter provides. Some commands can be *chained*—appended with other information, such as another entity name or a policy number.

Static items

Screen	Command
Account→Account Summary	Account <i>account number</i>
Account→New Submissions	NewSubmission
Administration tab	Admin
Administration→Business Settings→Activity Patterns	ActivitySearch
Administration→Users & Security→Organizations	OrganizationSearch
Search→Producer Codes	ProducerCodeSearch
Administration→Users & Security→Producer Codes	AdminProducerCodeSearch

Screen	Command
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	Policy <i>policy_number</i>
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→Business Settings→Policy Form Patterns	PolicyForms
Administration→Monitoring→Event Messages	Event Messages
Administration→Monitoring→Workflows	Workflows
Administration→Users & Security→Attributes	Attributes
Administration→Users & Security→Regions	Regions RegionSearch
Administration→Users & Security→Roles	Roles
Administration→Utilities→Script Parameters	ScriptParameters
Team tab	Team
Team→Activities	TeamActivities
Team→Other Policy Transactions	TeamPolicyChanges
Team→Renewals	TeamRenewals
Team→Submissions	TeamSubmissions
Team→Summary	TeamSummary
Server Tools→Batch Process Info	RunBatchProcess <i>process_name</i>

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	AccountRoles

Screen	Command
Billing	AccountBilling
Claims	AccountClaims
Account File Contacts	AccountContacts
Account File Documents	AccountDocuments
Account File Locations	AccountLocations
New Document – Create a new document from a template	AccountNewDocumentCreate
New Document – Link to an existing document on this account	AccountNewDocumentsLinkToExisting
New Note	AccountNewNotes
Account File Notes	AccountNotes
Account File Contacts→Role	AccountRoles
Submission Manager	AccountSubmissionManager
Account File Summary	AccountSummary
Underwriting File	AccountUnderwritingFiles
Account File Policy Transactions	AccountTransactions

Example:

```
Account account_number AccountRoles
```

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	PolicyAudit
Billing	PolicyBilling
Cancel Policy	CancelPolicy
Change Policy	ChangePolicy
Contacts	PolicyContacts
Documents	PolicyDocuments
History	PolicyHistory
Locations	PolicyLocations
New Document – Create a new document from a template	PolicyNewDocumentCreateNew
New Document – Link an existing document to this policy	PolicyNewDocumentsLinkToExisting
New Note	PolicyNewNote
Notes	PolicyNotes
Participants	PolicyParticipants
Payment	PolicyPayment
Policy Info	PolicyInfo

Screen	Command
Pre-Renewal Direction	PolicyPreRenewalDirection
Quote	PolicyQuote
Referral Reason	PolicyReferralReason
Risk Analysis	PolicyRiskAnalysis
Summary	PolicySummary
Policy Transactions	PolicyTransactions

Example:

```
Account account_number AccountRoles
```


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.

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 75.

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**.

Note: When entering a phone number as a free-text search criterion, enter a phone number appropriate for your default phone region and do not use an extension. If free text search finds a match, it returns the phone number formatted according to your default phone region.

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 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 basic 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 basic 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 the *Configuration Guide*.

Search type selection shortcuts

Search type selection shortcuts allow the user to enter a one-character or two-character string that directs the free-text search function to return only matches of a particular type. Search types with a corresponding shortcut string include exact match, prefix match, synonym match, and phonetic match. Adding the corresponding shortcut string to the end of a free-text search field entry activates the corresponding search type in the search algorithm.

The next example will contrast the results that the user obtains when not using a search type selection shortcut versus when using one. The user not inputting a search type selection shortcut enters a name such as "john" in the name field. The user then searches on the specified name. The results of this search can include matches of all types.

The search type selection shortcut feature allows the user to add a short string such as "-n" directly to the end of the search string, "john". The search string as modified is "john-n". Adding the shortcut restricts the results of the search. The results only contain matches of the type corresponding to the shortcut. In this case, the results would include only synonyms for "john". They would not include matches of any other type.

The following is a list of the search type selection shortcuts that Guidewire supports:

- -e : exact match only
- * : prefix match only
- -n : synonym (nickname) match only
- -p : phonetic match only

Basic search results for fields with multiple matches

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.

Note: PolicyCenter includes advanced search for additional object types. For more information, see “Advanced search” on page 75.

Basic search user interface

This topic describes the basic search user interface.

Policy basic search screen

Basic search provides search for policies.

Access policy basic search

Procedure

1. Navigate to **Search→Policies** and view the **Basic** tab.
2. 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 71.	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	Exact Query

Field	Description	Matching Type
	<ul style="list-style-type: none"> • (650)555-1234 • (650) 555-1234 • 650.555.1234 	
Official ID	Search for a Social Security number (SSN) or employer identification number (EIN) number.	Exact Query
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 basic search and ranking” on page 68.

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 basic search fields” on page 68.

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.

Field	Description
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.
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 search with in force on” on page 71.

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 search with in force on

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 two years ago. He renews the policy every year but makes no other changes. To view the policy in force on October 1 of this year, 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 of this year through January 1 of the following year. Jane married and changed her name to Jane Smith on July 1. You search for either Jane Doe or Jane Smith with an in force date of August 1 of this year. The search returns Jane’s policy that is in force on August 1 of this

year. If you search for Jane Smith with an in force date of August 1 of the previous year, 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 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 703 and "Turn on producer code security" on page 711.

Use basic policy search

About this task

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

Procedure

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 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**.

Basic search index updates automatically

About this task

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 "Use basic policy search" on page 72.

Procedure

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.

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.

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 67.

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 67.

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 702
- *Configuration Guide*

Minimum search requirements for advanced search

This topic describes the minimal requirements for advanced search 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 77
- “Assign submissions, renewals, and other policy transactions” on page 727 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 “Search for a contact” on page 358.

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 53.

Working with the Advanced Search tab

The following topic describes how to work with the **Advanced Search** tab in PolicyCenter.

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

About this task

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.

Procedure

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 for advanced search” on page 76.

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.

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 Easter 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

- “Assign submissions, renewals, and other policy transactions” on page 727 for a similar feature on the **Team** tab.

Search quotes user interface

The **Search→Quotes** screen enables you to search for quotes in the quote store and unbound submissions on the current system created from quotes in the quote store. To exclude the current system from search results, select **Only search Quote Store**.

Search requires that you specify at least one of the following:

- **Quote ID**
- **First Name and Last Name**
- **Work Phone**
- **Email Address**
- **Company Name**

See also

- *Configuration Guide*

Submission number

In **Search Results**, quotes from the quote store have a **Convert to Submission** link in the **Submission Number** column. When you click **Convert to Submission**, you must link the submission to an account on the **Account for Submission** screen. This screen is populated with account holder contact information from the quote. Use this information to search for an existing account or to create a new account. You can link to any existing account, even if it does not appear in the search results.

After attaching the account, the quote is copied to the current system as a submission with quoted status.

Quote ID

Quote-only instances generate the quote ID. In the base configuration, the **Quote ID** appears only in the **Search Quotes** screen. When you convert to submission, the quote ID is saved on the policy period (`PolicyPeriod.QuoteIdentifier`).

Saving your work

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

Unused work list

You can access your unused work from the **Unused Work**  list 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 unused 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 **Unused Work** list, as seen in the following example. Selecting it returns you to that screen with your unused data. You can finish your work and save it by clicking **Update**.

The **Unused Work** list 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 **Unused Work** list. However, if you attempt to log out without saving, PolicyCenter alerts you that your unused work will be lost if you continue.

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

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 page that appears on top of another page. It is part of the wizard process, but it is not a wizard step.

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 **Unused Work** list 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** list.

Part 3

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.

See also

- “Policy file” on page 293

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. 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.

Submission

Submission is the only policy transaction that creates a policy. A potential policyholder contacts the insurer 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 89

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 97

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 101

Cancellation

The cancellation process is a type of policy change which marks a policy as canceled. A cancellation can be initiated by the insurer. A cancellation initiated by the insurer typically requires advance notice to the policyholder.

Therefore, the insurer 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 insurer. According to the policyholder's wishes, the insurer cancels the policy effective immediately or at some future date.

See also

- "Cancellation policy transaction" on page 111

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 119

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 127

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 133

Audit

The audit policy transaction lets the insurer 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 139

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	New 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	•					•	•		•

Common Feature	Submission	Issuance	Policy Change	Cancellation	Reinstatement	Rewrite	New Account	Renewal	Audit
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	•	•	•	•	•	•	•	•	

See also

- “Underwriting issues” on page 671
- “Policy holds administration” on page 729

Preempting policy transactions

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 474

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 472

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	Reinstate-ment	Rewrite	Rewrite new account	Renewal	Audit
Submission									
Issuance									
Policy Change	•		•	•	•	•	•		•
Cancellation	•		•	•	•	•	•		•
Reinstatement	•		•	•	•	•	•		•
Rewrite	•		•	•	•	•	•		•
Rewrite New Account	•		•	•	•	•	•		
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. 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 445.

See also

- *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. 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.

See also

- *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 305 to learn more about accounts.
- “Issuance policy transaction” on page 97 to learn about issuing submissions as a separate policy transaction.

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 an insurer 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 a submission 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 insurer 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 information 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 in a submission

Pre-qualification questions are a screening tool used to determine applicant risk and to reflect the insurer'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 insurer 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 for 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

- “Underwriting issues” on page 671
- *Product Model Guide*
- *Configuration Guide*

Selecting an underwriting company in a submission

Large insurers 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 insurers to have more than one set of rates per underwriting company. Therefore, if an insurer 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 insurer may segment accounts into three different groups: high hazard, medium hazard, and low hazard. The insurer 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.

See also

- *Configuration Guide*

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 insurer 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 issues” on page 671
- *Configuration Guide*

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 and issuing a policy in a submission

PolicyCenter makes a distinction between binding and issuing a submission.

Binding means that the insured and the insurer 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 insurer 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 97 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

- *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

- “Copy a submission” on page 96 for an example of how to copy a submission
- “Multi-version quoting” on page 161
- *Configuration Guide*

Submission manager

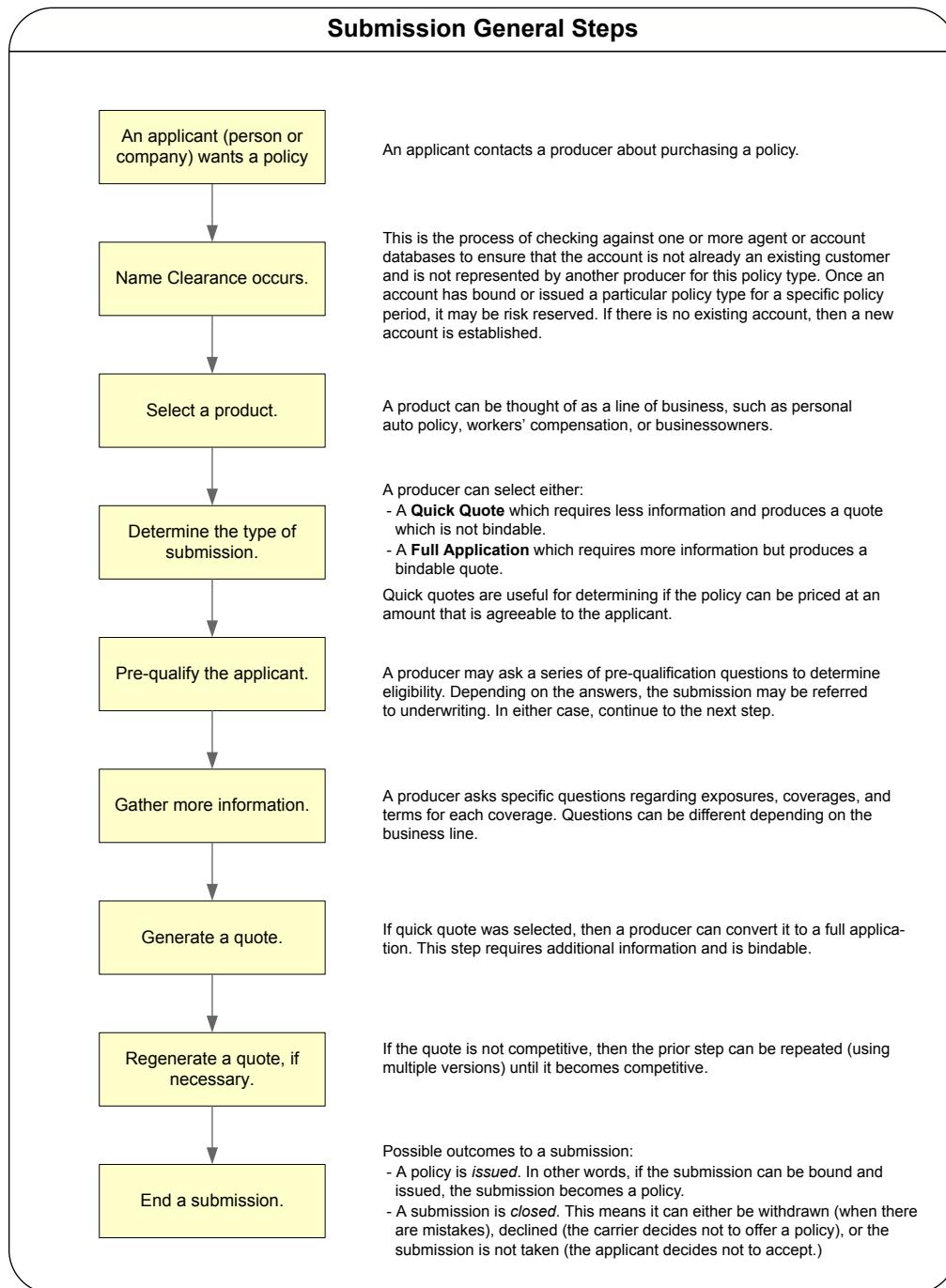
The **Submission Manager** screen contains summary information such as line of business, quote type, effective date, status of the transaction, and the premium. You can access this screen from the sidebar of an account. You can use this screen to do actions, such as withdrawing a submission. Use the **Submission Manager** 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 the submission. This drop-down list applies to both Quick Quote and Full Application and appears if the submission has not been bound. See “Closing a submission” on page 92 for additional information on these actions.

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.



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.

See also

- “Tools menu in policy file” on page 296
- “Working with multi-version quoting” on page 161

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 in these ways:

- **Actions→Create→New Submission** menu when viewing the **Desktop** and **Account** tabs and **Submission Manager** screen
- **Policy** tab, select **New Submission** from the drop-down list
- **Actions→Copy Submission** when viewing a submission or policy
- **Actions→Spin-off Policy from this One** when viewing a policy
- **Actions→Split Policy into Two** when viewing a policy

Create a submission

About this task

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 305.

Procedure

1. If an account already exists, you can begin a submission by selecting the account from the **Account** tab or from the **Desktop** tab, select **My Accounts** in the Sidebar.
2. While viewing the account, click **Actions→New Submission**. Since you have already defined an account, PolicyCenter directs you to the **New Submissions** screen which has default values for organization, 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 base application, you can create up to five per line of business. You can configure this maximum in Studio. In this example, you create only one submission, so click **Single**.
4. In **Quote Type**, select the type of quote you want (**Quick Quote** or **Full Application**). In this example, select **Full Application**.
5. Choose from the available product offerings.
 - **Quick Quote** gathers the minimal information needed to generate a quote.
 - **Full Application** gathers complete information needed to bind and quote.
6. 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.
7. The **Policy Info** screen allows you to collect information, determine policy details
If you have the correct permissions, you can 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.
8. Continue entering required information in the submission wizard. Each line of business has 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 273 for more information on workers’ compensation.
 - See “[Personal auto](#)” on page 257 for more information on personal auto.
 - See “[Businessowners](#)” on page 167 for more information on businessowners.
9. After entering the required information, you can review it in the **Policy Review** screen. If you are satisfied, then click **Quote**.

10. 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.
11. *(Optional)* Select **Forms** to view the list of forms that will be attached to the policy.
12. Select **Payment**. Enter the type of billing plan, payment type, and the deposit collected on the **Payment** screen.
13. Select a bind option from **Bind Options**:
 - **Bind Only**, legally binds both the insurer and the applicant, generates billing information, but does not issue the policy.
 - **Issue Policy**, binds, generates billing information, and issues the policy.

Copy a submission

About this task

You can copy information from a submission to create a new submission. For additional information on this see “Copying submission information” on page 93.

Procedure

1. Navigate to a policy or submission.
2. From the **Actions** menu, select **Copy Submission**.
3. Make changes to the policy.
4. As with any submission, 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 insurer 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. 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.

See also

- *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 insurer has agreed to provide coverage, there are some details that must be confirmed before generating the paperwork.

How issuance works

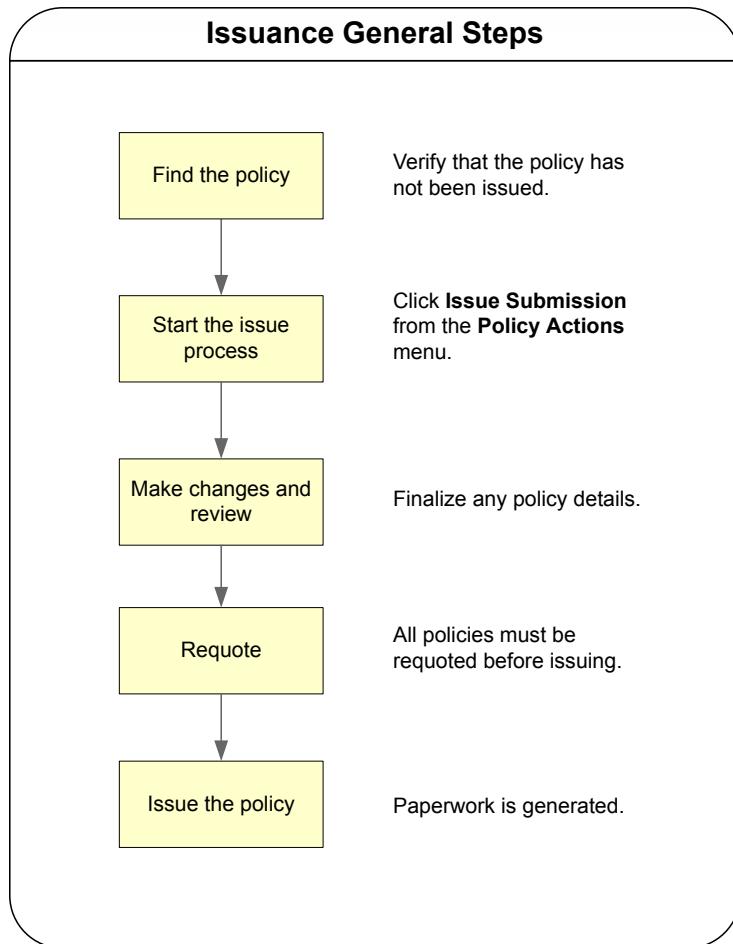
To start an issuance policy transaction, the policy must already be bound, but not issued. In addition, the policy must be no other open policy transactions. The issuance wizard behaves in much the same way as the submission wizard. You can edit in the **Policy Info** screen, review the policy, requote, and change payment options. To issue a policy, you must have the **bindissuance** system permission. An insurer 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 issuance

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

See also

- “Tools menu in policy file” on page 296

Issue a policy

About this task

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

Procedure

1. Navigate to the **Policy Summary** screen, or any screen in the policy.
2. From the **Actions** menu, select **Issue Policy**.
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. Click **Quote** to requote the policy.
5. Click **Issue Policy**. A dialog box asks you to verify your action. In a production system, PolicyCenter might send 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. The activity is assigned to the underwriter.

Renewal policy transaction

Insurers 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.

Note: In PolicyCenter, the user interface uses the term policy transaction to refer to submissions, policy changes, and other policy transactions. 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.

See also

- *Configuration Guide*

Renewal overview

Typically, as a policy nears its expiration date, insurers 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 an insurer.
- Reduce expenses associated with the renewal process.

For both the producer and the insurer, renewing an existing customer is more profitable than acquiring a new customer with a similar profile because of acquisition and processing costs. An insurer'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
- Insurer 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.

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.

Starting renewals 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

- *Configuration Guide*

Starting renewals 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

- *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

- 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 insurer decides not to renew the policy, and the policy expires on the expiration date.

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.

Referral reasons on renewals

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:

- “Add underwriting referral reasons” on page 674
- “Underwriting referral reasons raise underwriting issues” on page 652
- *Configuration Guide*

Underwriting issues in renewals

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.
- **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.

Policy changes and 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. This is not preemption.

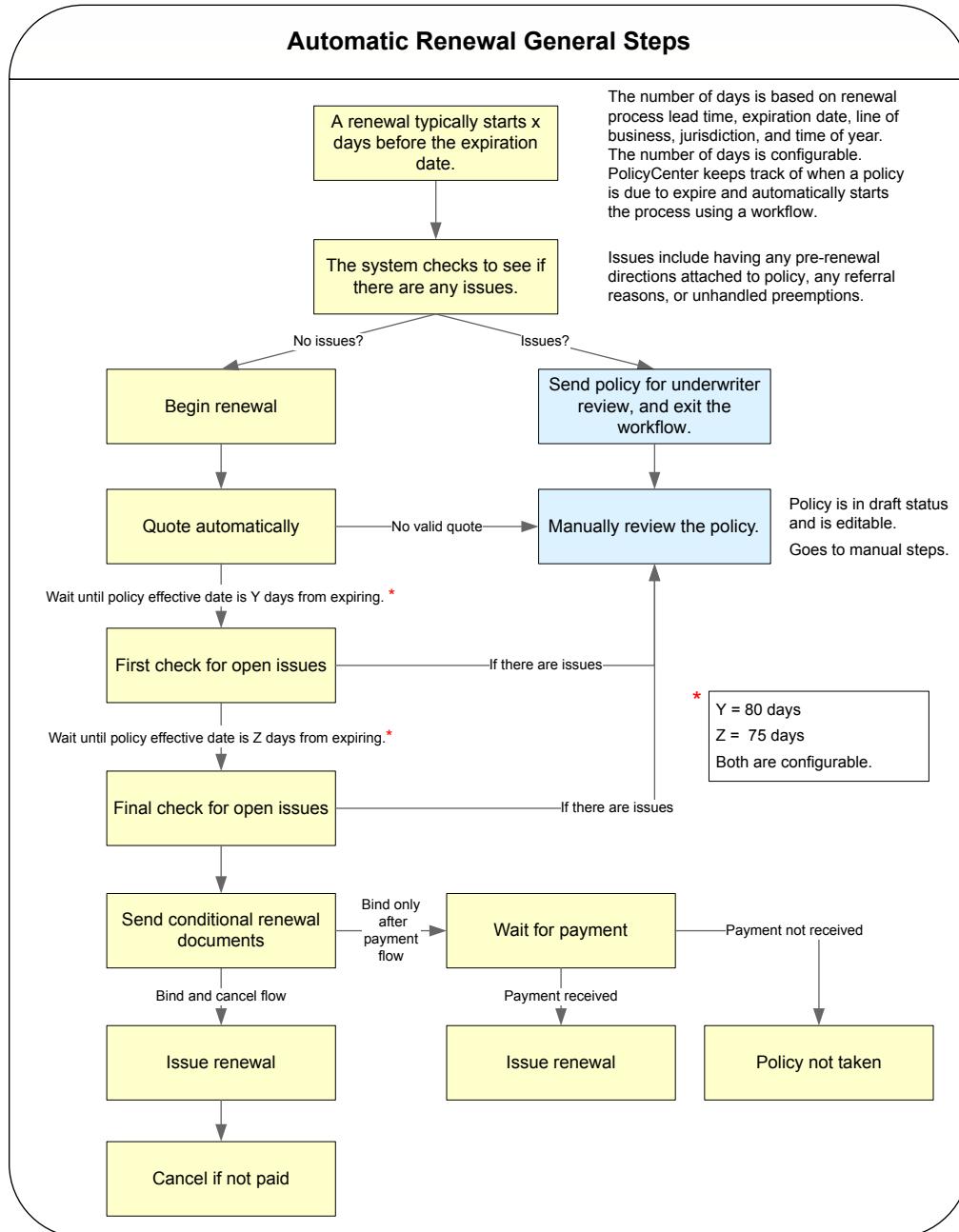
See also

- “Applying changes to future renewals” on page 477

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.

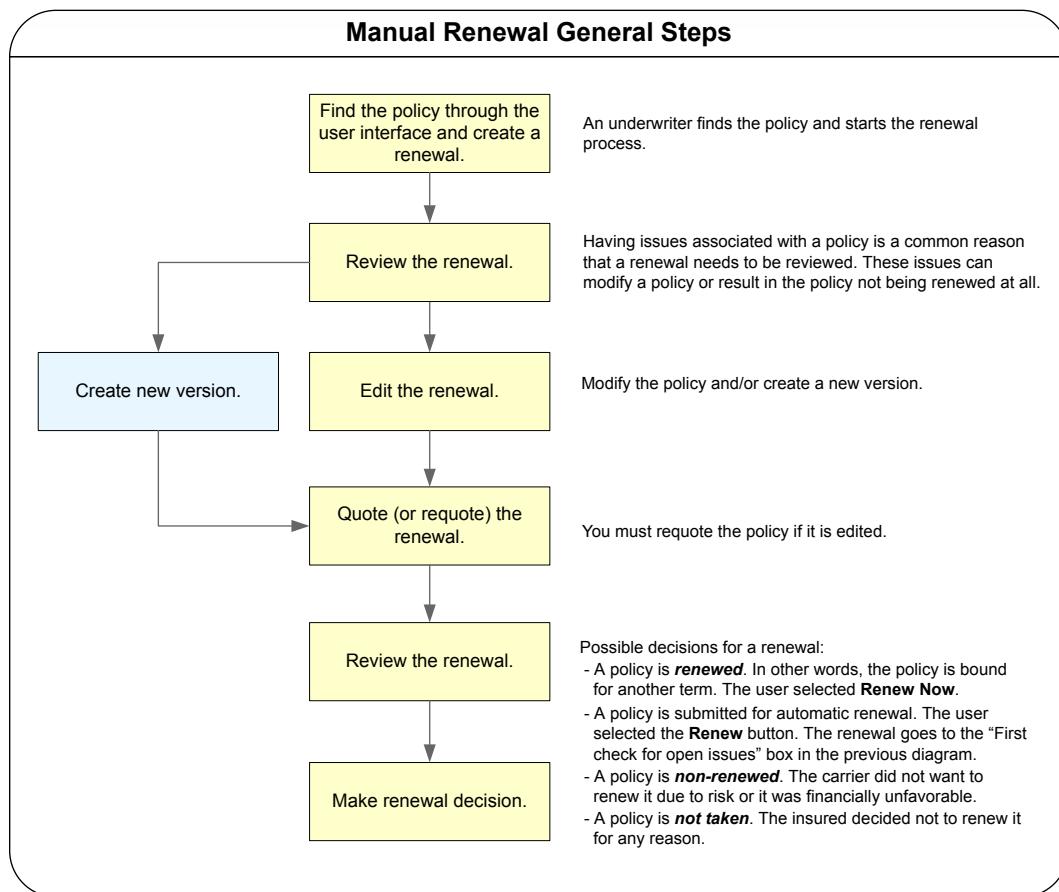
PolicyCenter uses an automatic process to renew a policy without human intervention. To view the default steps, see the *Configuration Guide*.



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.



Working with renewals

These topics explain how to work with renewals from a user's point of view.

See also

- “Tools menu in policy file” on page 296
- “Working with multi-version quoting” on page 161

Create a manual renewal

About this task

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

Procedure

1. Navigate to the policy you wish to renew.
2. From the **Policy Summary** screen (or any screen in the policy), select **Actions**→**Renew Policy**. A dialog box asks you to confirm your selection and the renewal wizard begins.
3. Advance to the **Policy Info** screen.
4. Click **Edit Policy Transaction**. Make the required changes to the policy.

5. 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.
6. For this example, click **Quote**.
7. On the **View Quote** screen, first review the quote and the policy in general.
8. Now you have various options for binding or closing the policy.
9. Renew the policy by selecting **Bind Options→Renew**.
10. Select a **Renewal Code** on the **Renewal Data Entry** popup:
 - <none>
 - **Renew - account consideration**
 - **Renew - assigned risk**
 - **Renew - good risk**
 - **Renew - legal requirement**
 - **Renew - producer consideration**

You can use this field to document renewal exceptions. For example, all values except Good Risk indicate that the policy is being renewed because of market or statutory compulsion. The default would typically be Good Risk. You can configure the **Renewal Data Entry** typelist values to meet specific needs or customize reason codes.

Create a renewal from a batch process

About this task

Creating a renewal from a batch process is the most common way for renewals to be created. The renewal batch process looks for policies that expire in X days (the default is 180 days). Then the Policy Renewal plugin further 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 **View BatchProcess tools page** permission. The code for this permission is `toolsBatchProcessview`. The Tools View role has this permission.

Procedure

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

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, day of month, month, and day of week.

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 **RenewalProcessLeadTime** 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.
- *Configuration Guide*

View your renewals

About this task

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.

Procedure

1. Go to the **Desktop** and select **My Renewals**.
2. You can filter your search by selecting from the drop-down menu.

Create multiple versions of a renewal

About this task

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.

Procedure

1. Select **New Version** in the renewal wizard to create a new version of the renewal where you can make changes and obtain a different quote.
2. 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

- “Create a manual renewal” on page 107 for an example.

Create a pre-renewal direction

About this task

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

Procedure

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 and set the security level which controls who can view it.
In the base 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
5. If you selected a non-renew direction, click **Add** in **Selected Non-Renewal Explanations** to add a non-renewal explanation.

See also

- *Configuration Guide*

View, edit, or delete an existing pre-renewal direction

Before you begin

“Create a pre-renewal direction” on page 109

About this task

You can view, edit, or delete a pre-renewal direction on a policy.

Procedure

1. In this example, navigate to a policy that has a pre-renewal direction.
2. Click the **Summary** link under the **Tools** menu to display the **Policy Summary** screen.
In the **Details** listview, a link to the pre-renewal direction appears under **This policy has pre-renewal direction**.
3. Click the pre-renewal direction link view the pre-renewal direction on the **Pre-Renewal Direction for Policy Term** screen.
From this screen you can **Edit**, **Delete**, or **View Notes** on the pre-renewal direction.

Cancellation policy transaction

A cancellation policy transaction is the process of voiding a policy while it is in force. Initiated either by the insurer 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 insurer 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 insurer rescinds the cancellation, and PolicyCenter withdraws the cancellation.

Another example of cancellation is when the insurer cancels the policy. The insurer 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 insurer 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 127.

Note: In PolicyCenter, the user interface uses the term policy transaction to refer to submissions, policy changes, and other policy transactions. 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.

See also

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

Scheduling a cancellation 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

- *Configuration Guide*

Default cancellation effective date

The insurer 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 insurer 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 insurer, 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 can never override the cancellation effective date for a flat cancellation because the whole policy term has been canceled. The user must have the **Cancellation override effective date** permission. The code for this permission is `cancelovereffdate`.

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*

Source	Reason	Refund method	Default cancellation effective 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.
Insurer	Cancellation of underlying insurance	Pro rata	Calculated based on jurisdiction and line of business.*
Insurer	Condemned/unsafe	Pro rata	Calculated based on jurisdiction and line of business.*
Insurer	Non payment	Pro rata	Calculated based on jurisdiction and line of business.*
Insurer	Policy rewritten (mid-term)	Pro rata	System's current date*
Insurer	Policy rewritten or replaced (flat cancel)	Flat	The date the policy went into effect.
Insurer	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

- *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 127.

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 have a policy that has an open cancellation for non-payment. The insured calls and asks to cancel the policy 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.

Canceling 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 insurer 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 insurer decides to give an extension on scheduled cancellations for all policies in or near the catastrophe. The insurer 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. The code for this permission is `canc1reschedule`. In the base configuration, underwriters and underwriter supervisors 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 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 127.

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 with pending renewals

If you cancel a policy that has a future renewal, the renewal is handled in various ways depending upon whether the renewal is bound.

Renewal is bound

If the policy has a future renewal that is bound, start a cancellation for the bound renewal term.

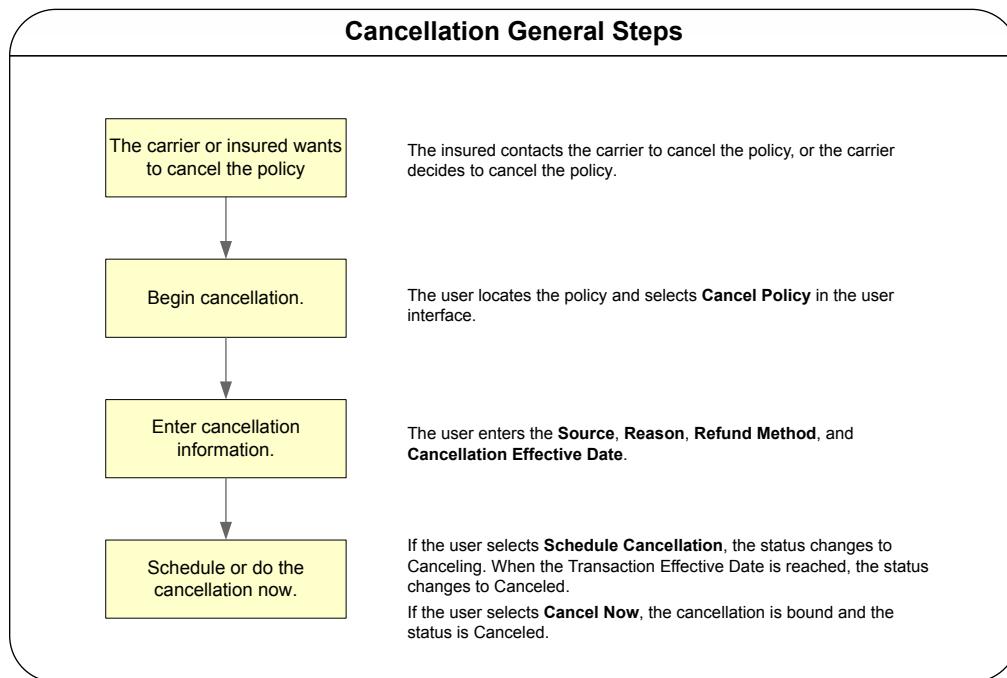
Renewal is unbound

If the policy has a future renewal that is not bound, the renewal is handled in the following ways:

- If renewal documents have not been sent, then withdraw the renewal. This removes the renewal job and all policy periods associated with the renewal.
- If renewal documents have been sent and there is enough time to send non-renewal documents, then withdraw the renewal. This withdraws the selected version of the policy period and blocks other policy periods in the renewal job through an underwriting issue.
- If renewal documents have been sent and there is not enough time to send non-renewal documents, then block the renewal. This blocks all policy periods in the job with an underwriting issue.

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

These topics explain, from a user's point of view, how to work with cancellations in the user interface.

See also

- “Tools menu in policy file” on page 296

Cancel a policy

Procedure

1. Navigate to a policy and select **Cancel Policy** from the **Actions** menu.
2. Enter the following information:
 - a. For **Source**, select **Insured** or **Insurer**.

- 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.
Certain source and reason choices result in a default refund method.
 - 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.
You can configure this in `CancellationEnhancement.gsx` and in the `notificationconfigs.xml` system table. See the *Configuration Guide*.
3. Click **Start Cancellation**.
 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 111. If you select **Schedule Cancellation**, then PolicyCenter might notify the document production system to prepare and mail 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**.
If the policy cancellation is scheduled but has not gone past its cancellation process date, you can rescind the policy.

Rescind a cancellation

About this task

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:

Procedure

1. Navigate to a policy with a cancellation pending.
Policies awaiting cancellation display **The Policy is Pending Cancellation** on the **Policy Summary** screen.
2. From the **Actions** menu, select **Rescind Cancellation** then select the cancellation.
The **Confirmation** screen appears and the status is **Canceling**.
3. Select **Close Options**→**Rescind Cancellation**.
The policy remains in force. You can view the rescinded cancellation in the **Policy Transactions** screen.

Change the cancellation effective date in a quoted cancellation

Before you begin

Follow the steps in “Cancel a policy” on page 115 until you arrive at the **Confirmation** screen.

About this task

You must have the Cancellation reschedule (`cancelreschedule`) permission to change the cancellation effective date in an open cancellation.

Procedure

1. Click **Back** to return to the **Entry** screen.
2. Click **Edit Policy Transaction**.

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

Change the cancellation effective date in a scheduled cancellation

Before you begin

Follow the steps in “Cancel a policy” on page 115 and select to schedule a cancellation in the future.

About this task

You must have the Cancellation reschedule (`cancelreschedule`) permission to change the cancellation effective date in an open cancellation.

Procedure

1. Navigate to a policy with a cancellation pending.
Policies awaiting cancellation display a message that the policy is pending cancellation. The policy **Summary** screen displays **Pending Policy Transactions**.
2. Click the **Transaction #** link for the cancellation.
3. Click **Edit Policy Transaction** on the **Confirmation** screen.

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

Policy change transaction

A policy change transaction is a modification made to a policy while it is in-force. The types of changes that might happen to a policy mid-term include:

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. 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.

See also

- *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 123 and “Using preemption in a policy change” on page 123 for more information.

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:

- Changes you have made
- Conflicts that you need to resolve before binding (if the policy change is out-of-sequence)
- Conflicts related to preemption

Although the **Policy Review** screen displays changes to the policy, view the **Quote** screen to see how your changes affect the premium:

- 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.

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, “Underwriting issues and editing the effective date in policy change” on page 122.

Multi-version policy change

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.

New effective date in policy change 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.

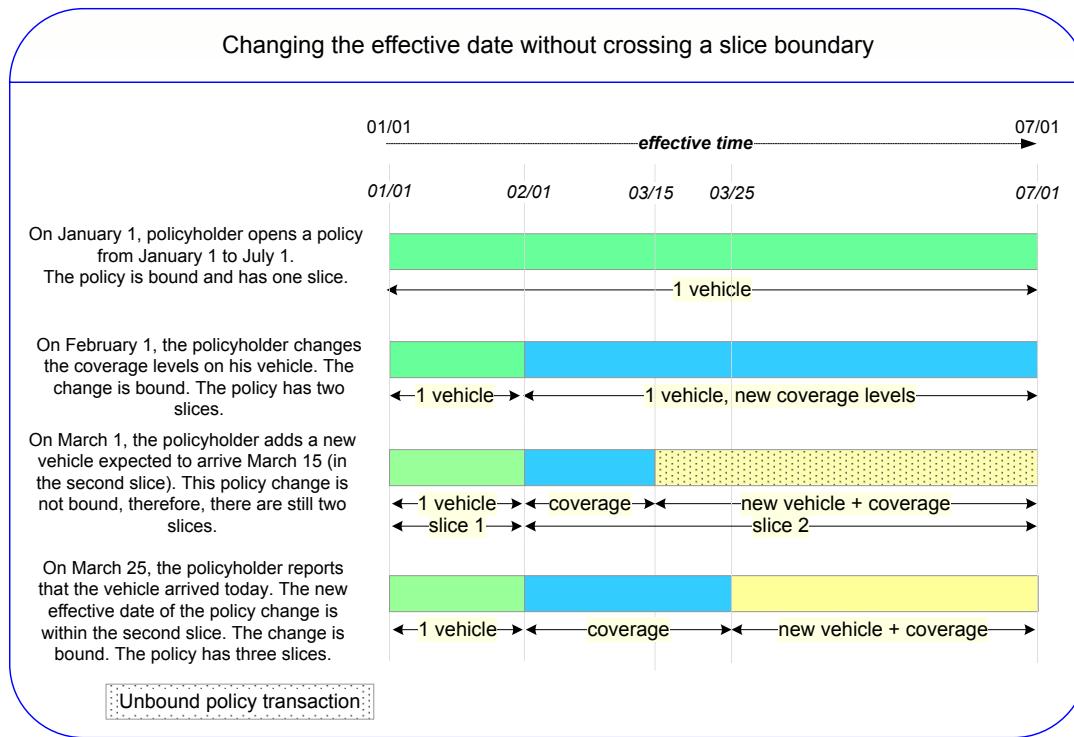
For more information about slices, see “Slice mode and window mode overview” on page 462.

Example that does not cross slice boundaries

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

- 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.
- 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.
- 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.
- 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

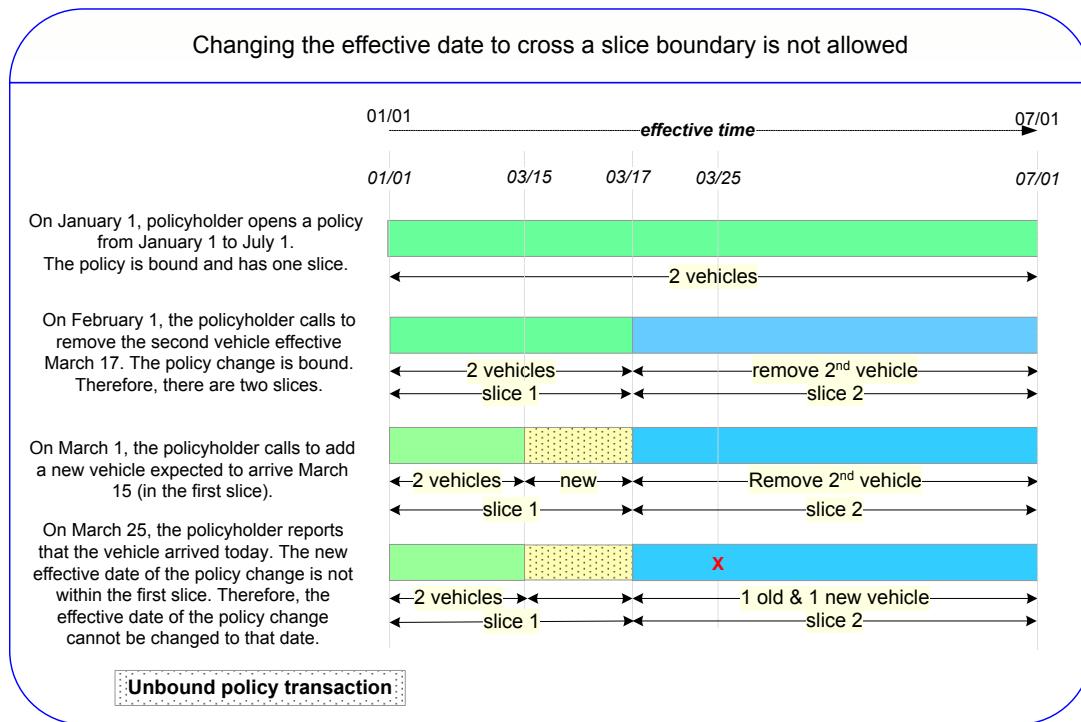
The following example crosses slice boundaries. Therefore, it is not allowed.

- 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.
- 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.
- 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.
- 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.



Underwriting issues and editing the effective date in policy change

In general, when you edit the effective date of a policy change, PolicyCenter preserves existing underwriting issues and approvals. There is some variance in that way that 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. Insurers 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 472

Using preemption in a policy change

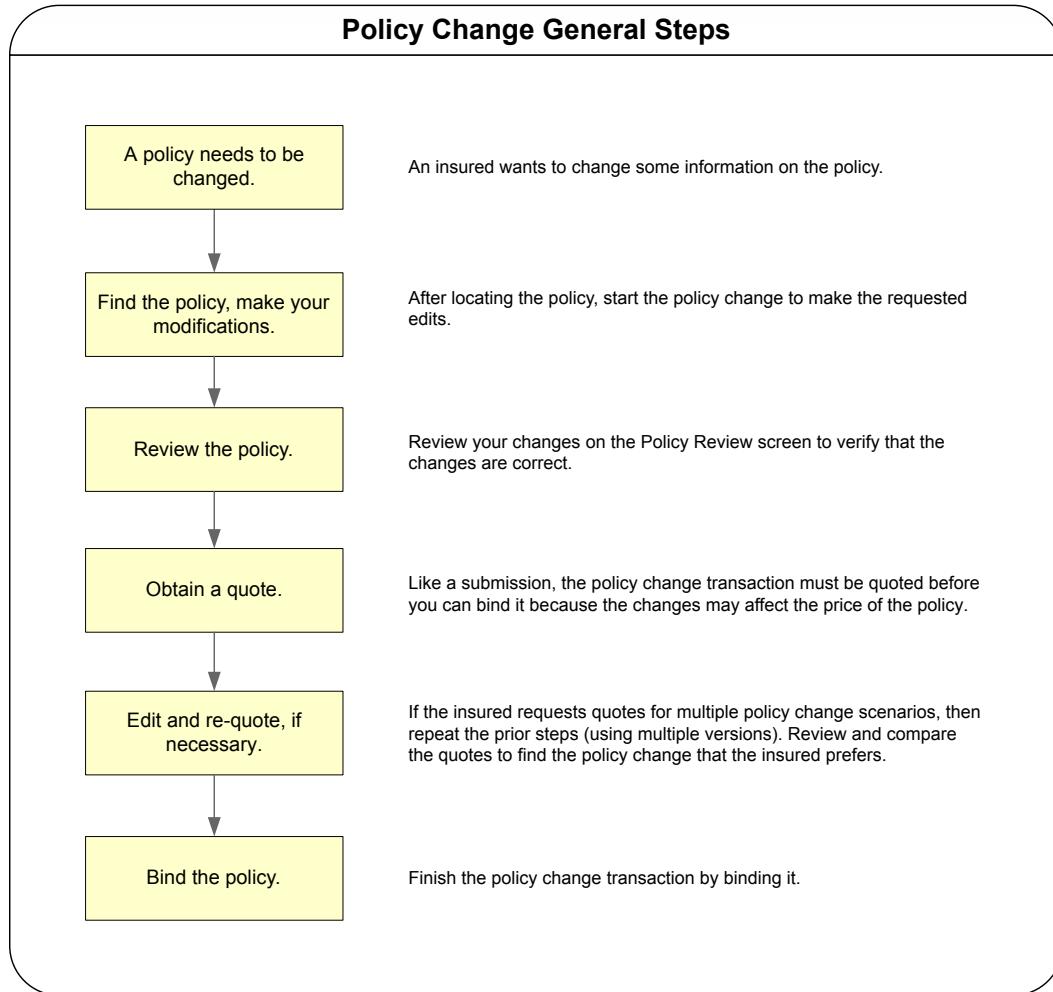
A preemption can occur if two policy transactions are open on a policy at the same time. The first policy transaction that binds preempts the second one. PolicyCenter handles preemptions by merging the changes made in the preempting policy transaction with the changes made in the preempted policy transaction. 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 474

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.



Working with policy changes

IMPORTANT Since the policy change process for any insurer 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 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

- “Tools menu in policy file” on page 296
- “Working with multi-version quoting” on page 161

Create policy change

Procedure

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.

Although 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. The 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?”

- 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.

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.

Edit the policy change effective date

About this task

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

Procedure

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 in policy change must be within the same slice and policy term” on page 120.
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.

Change the policy expiration date in a policy change

About this task

Follow these instructions if you need to change the policy expiration date in a policy change transaction.

Procedure

1. Navigate to the **Policy Info** screen.
2. In **Policy Details**, select **Other** from the **Term Type** drop-down menu to make the **Expiration Date** field editable.

Reinstatement policy transactions

Reinstatements 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. 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.

See also

- *Configuration Guide*

Reinstatement overview

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

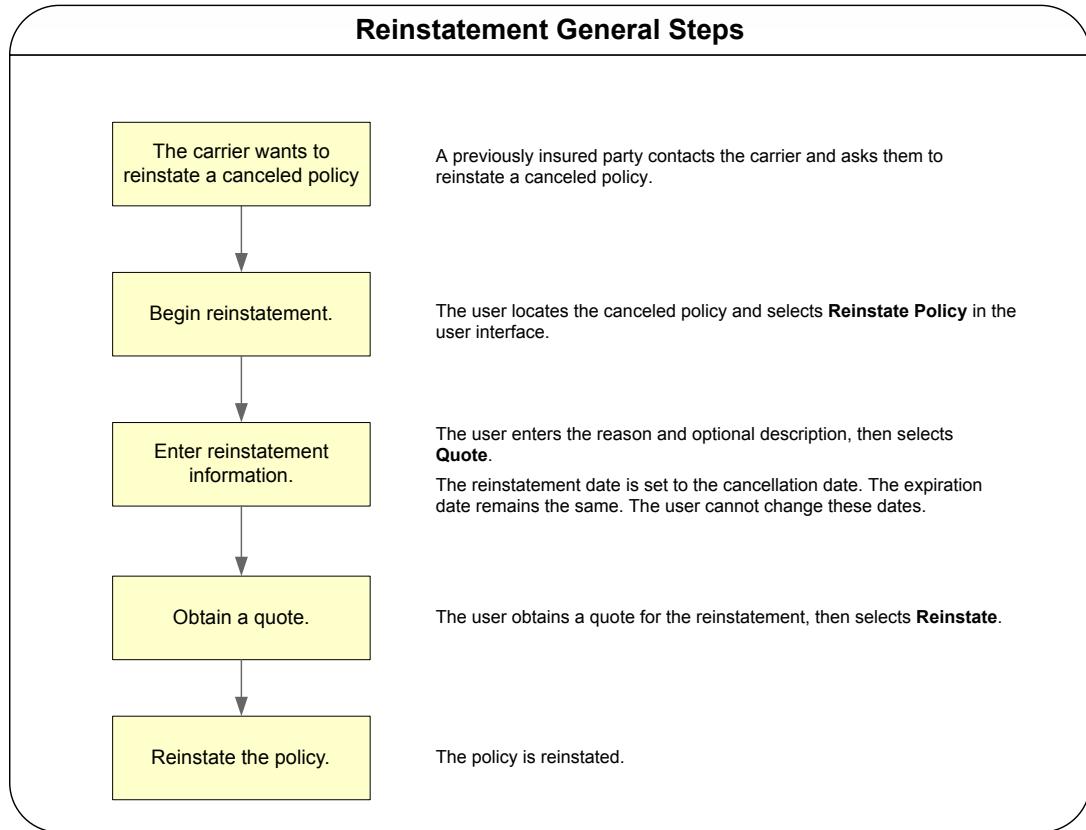
From the policyholder’s perspective, a reinstated policy is no different than the original policy. However, PolicyCenter tracks the reinstatement as a policy transaction.

Reinstatement has the following features:

- You can reinstate a canceled policy.
- The reinstated policy cannot have a lapse in coverage.
- You cannot make changes to the policy in a reinstatement.
- You cannot reinstate a canceled policy with a new expiration date.

Reinstatement general steps

The following diagram shows the basic steps in a reinstatement. These steps are for the default configuration of PolicyCenter. Your 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.



Working with reinstatements

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

Reinstate a policy

About this task

You can start a reinstatement from a canceled policy.

Procedure

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

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

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

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

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

- a. Select **Edit** to edit the **Reason Description** field. You must select **Quote** to return the policy to quoted status. However, the amount in the quote does not change.
- b. Select **Reinstate** to reinstate the policy.
- c. Select **Withdraw Transaction** to stop the process. The policy remains in canceled status. You can decide at a later date to reinstate the policy.
- d. Select **Print Quote** to print the quote.

Rewrite policy transactions

Insurers must do a rewrite if they need to change the effective date of the policy or producer of record. Insurers 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.

Insurers 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 insurer 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 insurer 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 insurer makes this change as a mid-term rewrite to simplify tracking of this change for both the insurer and the insured. Rewrite reissues the policy documentation rather than sending an addendum, and the insurer 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. 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.

See also

- *Configuration Guide*

Rewrite overview

In the base configuration, 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, rewrite can be configured independently of submission, 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 both the newly rewritten policy and the original policy version still exist in PolicyCenter.

Change any policy information in a rewrite

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.

Full-term rewrite

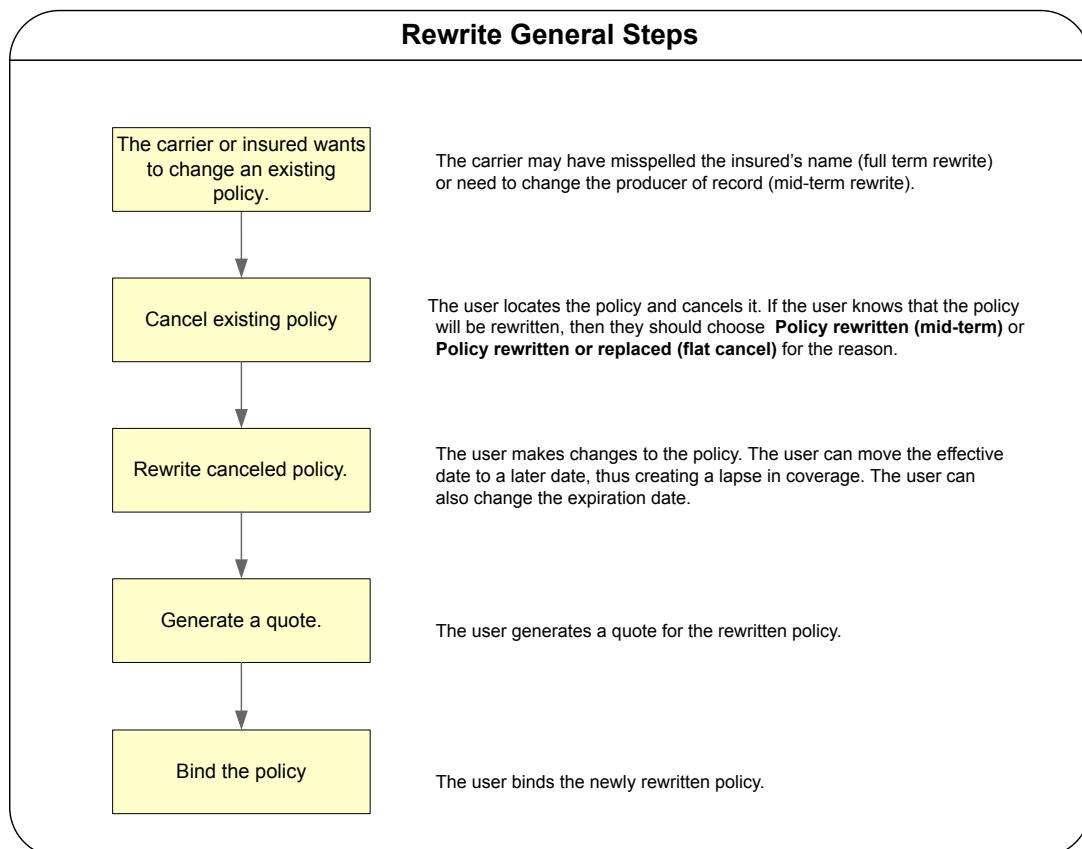
A full-term rewrite replaces the original policy for the complete policy term. A full-term rewrite can have a lapse in coverage.

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.

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.



Working with rewrites

These topics explain, from a user's point of view, how to rewrite a policy.

Rewrite a policy

About this task

The rewrite process starts from a canceled policy.

Procedure

1. Navigate to a 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.

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.

Note: To create a lapse in coverage, change the **Effective Date** to a later date.

3. Make your changes to the policy, and then select **Quote**.
4. Select **Issue Policy** to issue the policy.

Approve, request changes, or decline a rewrite

About this task

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.

Procedure

The underwriter reviews the rewrite and may choose to approve it or edit it then requote the policy.

- The underwriter selects **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.

View rewritten policies

About this task

Rewriting a policy creates a new version of the policy.

Procedure

You can view all versions of the policy by going to the Account and viewing the **Account File Summary** screen.

The **Policy Terms** listview shows the original policy that was canceled, and the full-term rewrite of the policy that is currently in force.

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. 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.

See also

- “Moving and rewriting policies between accounts” on page 306
- “Rewrite policies from one account to an existing account” on page 316
- “Overview of split and spin-off policies” on page 294
- *Configuration Guide*

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 rewrite new account started on source policy

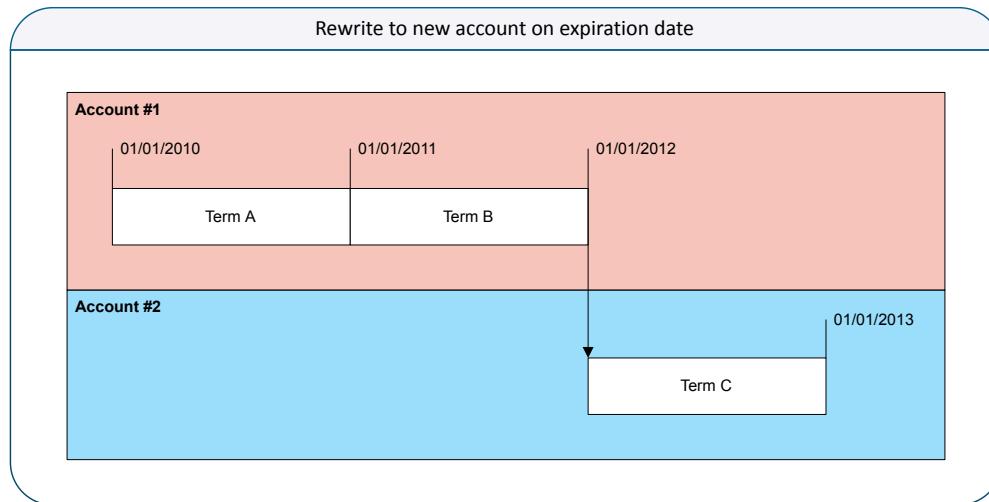
The rewrite new account policy transaction rewrites a source policy to a target policy. There are a number of restrictions that apply to a rewrite new account policy transaction. Primarily, the restrictions prevent the target policy from having an active term that overlaps with the source policy.

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

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.

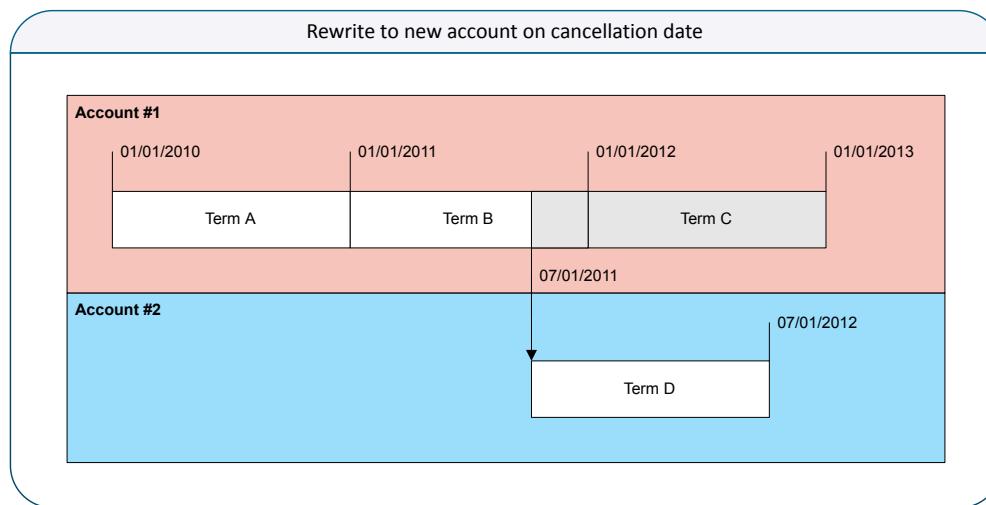
Policy transaction	Restriction
Submission	Not applicable.
Policy Change	Not allowed on and after the expiration date of the source policy.
Cancellation	Not allowed on and after the expiration date of the source policy. Does not differ from the usual behavior.

Policy transaction	Restriction
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.



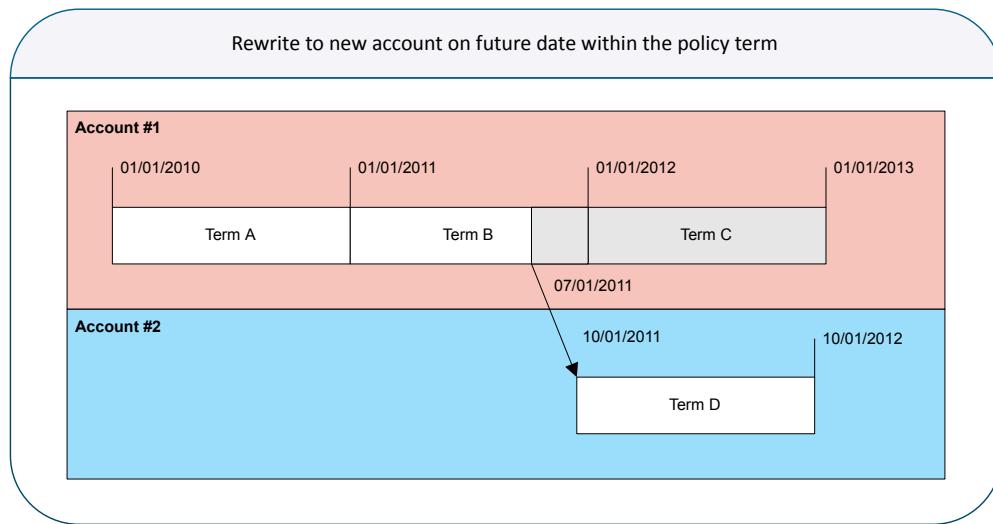
The following table shows restrictions for policy transactions on the source policy.

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.



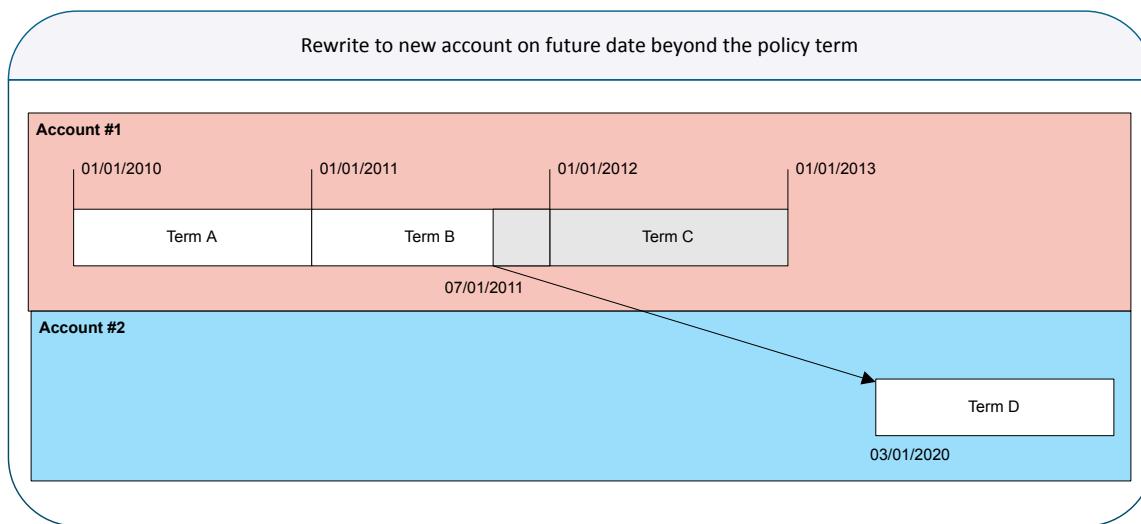
The following table shows restrictions for policy transactions on the source policy.

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 rewrite new account started on target policy

A rewrite new account 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 Policy Renewal Start 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.

Rewrite policy to new account

Before you begin

“Rewrite policies from one account to an existing account” on page 316

About this task

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.

Procedure

1. As the same user who rewrote the policy from one account to another, go to the **Account Summary** screen.
2. In **Current Activities**, click the **Rewrite to new account** activity generated by rewriting from one account to another. PolicyCenter jumps to the **Rewrite New Account** policy transaction. The bottom part of the window displays the **Activity Detail** tab.
3. 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.
4. Quote and issue the policy.
5. In the **Activity** tab at the bottom of the screen, click **Complete** to complete the activity.

Premium audit policy transaction

In the base configuration, PolicyCenter provides two types of premium audits: *final audit* and *premium report* policy transactions. You can configure these audit types to meet your requirements or add new configurable audit types. For information about configuration, see the *Configuration Guide*.

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 a policy which runs from January 1 of this year to January 1 of the following year 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 765.

Final audit overview

Note: In the base application, the workers’ compensation and general liability lines of business are set as auditable.

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 insurer. 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

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.

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.

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 insurer sends a document to the policyholder requesting the required information. The policyholder adds the information and returns the document to the insurer for processing. This document contains

instructions for the policy holder to provide actual exposures for the reporting period. Generally, this includes class codes and other rating information.

Where does the insurer send premium audit information?

Final audits are important to determine actual policy premium. The insurer 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 insurer. 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

- “Add a new audit” on page 149 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 insurer 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 insurer 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

- “Enter premium report data” on page 152 to view the trend analysis information in the application
- *Configuration Guide*

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

- “Add a new premium report” on page 152

Premium 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 audit roles as described in the following topics.

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 insurer’s office. The premium auditor role can vary by insurer. 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 insurer. 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 document from the policyholder. The examiner reviews the document 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.

Policy transaction	Description
	<p>Premium Report Policy Transaction</p> <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.
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> <p>Final Audit Policy Transaction</p> <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. <p>Premium Report Policy Transaction</p> <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	<p>Final Audit Policy Transaction</p> <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. <p>Premium Report Policy Transaction</p> <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	<p>Final Audit Policy Transaction</p> <p>When the rewrite finishes, PolicyCenter schedules a final audit for the new period.</p> <p>Premium Report Policy Transaction</p> <p>According to the configuration settings, PolicyCenter schedules premium reports when you issue the rewritten policy.</p>

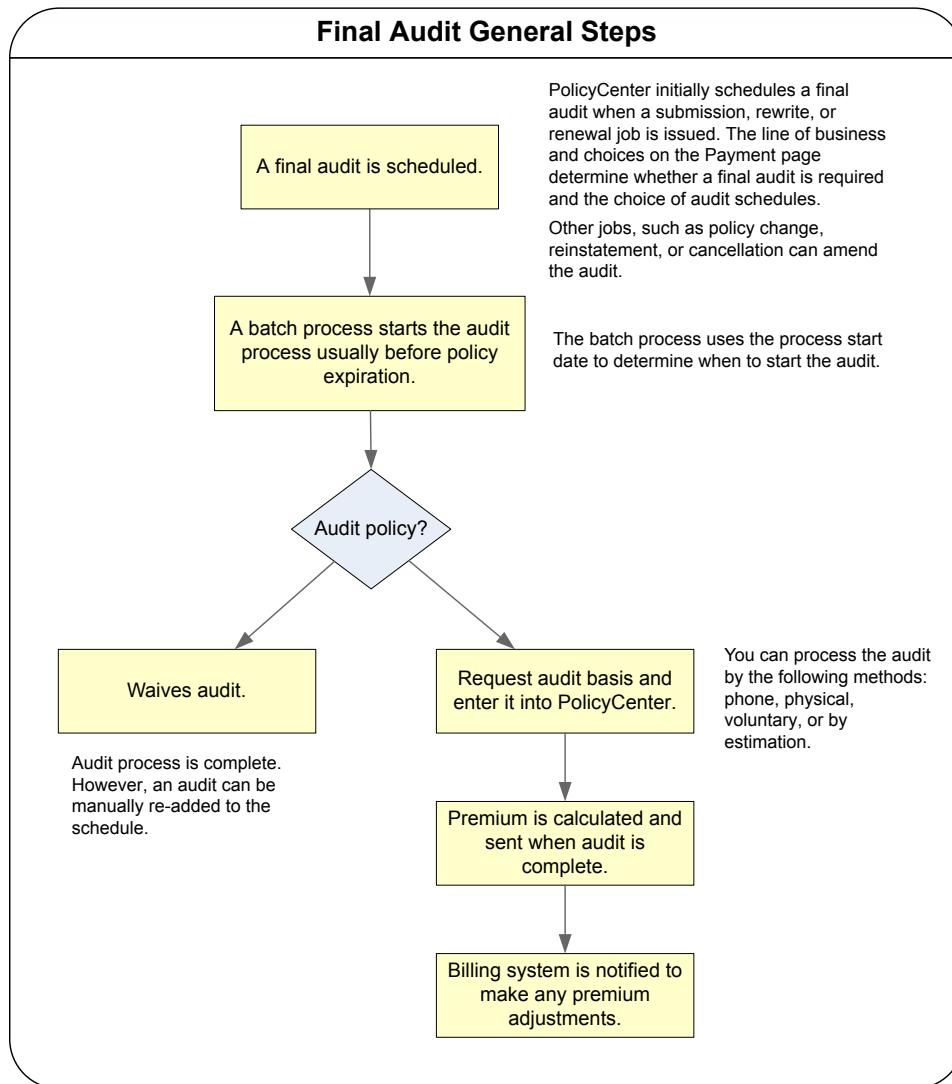
Note: Final audits cannot be out of sequence with other policy transactions. However, final audits can be preempted.

Premium audit general steps

This includes topics that describe the general steps for premium audit.

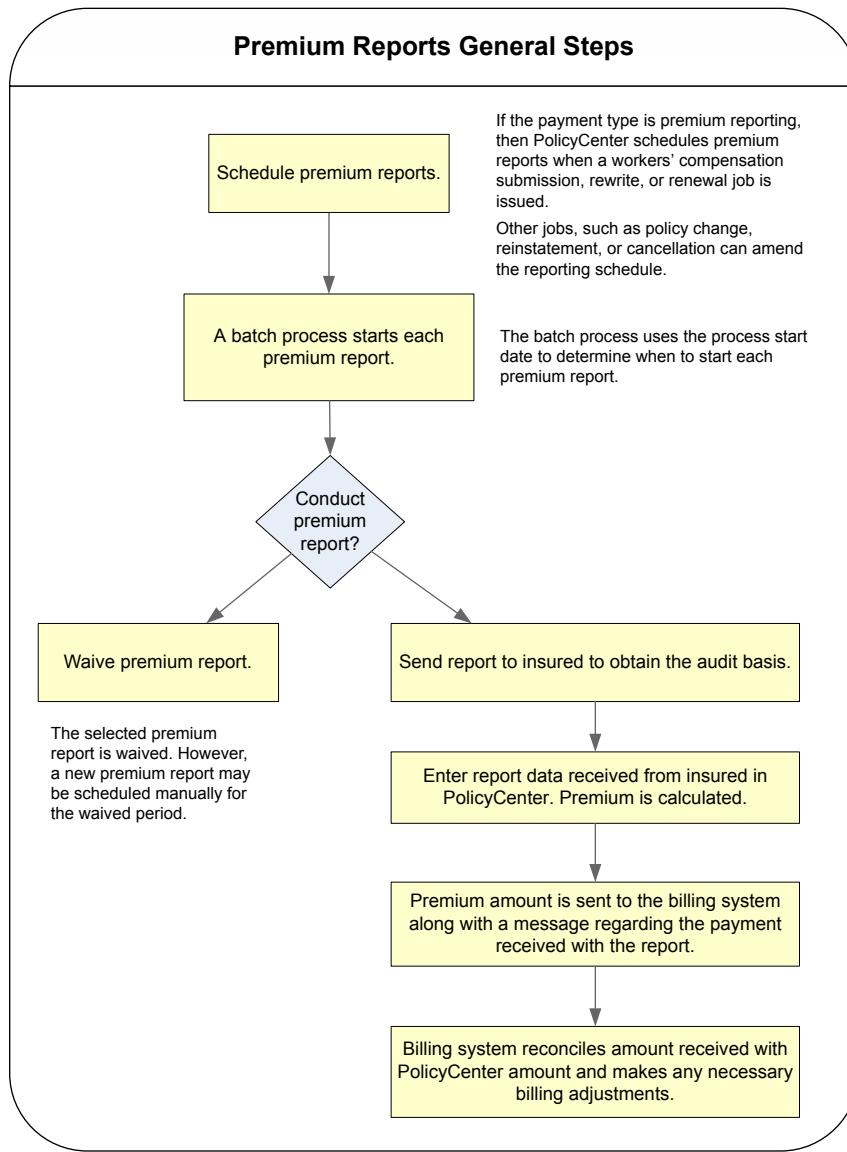
Final audit general steps

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 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.

The included topic provide step-by-step instructions.

See also

- “Tools menu in policy file” on page 296

View final audit schedule

About this task

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.

Procedure

1. Navigate to a policy that can be audited.
2. Select **Audit Schedule** in the **Tools** menu. The **Audit Schedule** 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.

View audit activities

About this task

The Audit Task 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.

Procedure

1. To find audit activities that are assigned to you, navigate to the **Desktop** tab and click **My 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.

Edit, start, or waive a final audit

Before you begin

“View final audit schedule” on page 148

About this task

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: To edit an audit, you must have the **Edit audit** (**editaudit**) permission. To start an audit, you must have the **Start audit job** (**startaudit**) permission. To waive an audit, you must have the **Waive audit** permission (**waiveaudit**).

Procedure

1. Navigate to the **Audit Schedule** screen for a policy.

2. Under the **Actions** column, select one of the following choices:

Edit Change the **Process Start** date, **Due Date**, or planned **Audit Method**.

Start Send the audit for processing by the Audit Task batch process.

Waive Bypass the audit procedure. PolicyCenter will not perform an audit on that audit period. If you later decide that you need the audit, you can add it back in. See “Add a new audit” on page 149.

When an audit is in progress, the **Edit** and **Waive** buttons are not available. Changes or waives must be done within the audit. See “Enter audit data and complete final audit” on page 149.

Add a new audit

About this task

In certain cases, you can add a new audit or premium report. “Adding a final audit” on page 141 describes when you can add a new audit. “Adding a premium report” on page 143 describes when you can add a new premium report.

Procedure

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 **OK**.

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 audit 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**.

Enter audit data and complete final audit

About this task

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: To edit an audit, you must have the **Edit audit** (`editaudit`) and **Quote** (`quote`) permissions. To complete an audit, you must have the **Complete audit** (`completeaudit`) permission.

Procedure

1. Perform the steps in “View final audit schedule” on page 148 or “View audit activities” on page 148. Then click the link of the **Final Audit** you wish to complete.
2. 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.

Field	Description
Instructions	Enter instructions for the premium auditor or premium audit examiner.

3. Click **Next**.
4. 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.
5. 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.
6. To see the calculation of the total premium for the audit term, select the **PremiumDetails** 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.
7. If you need to change the class code exposures, click **Edit Audit**.
8. When the amounts are accurate, click **Submit**. The audit status is now **Completed**.

Revise a final audit

About this task

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 the **Revise audit (reviseaudit)** permission.

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:

Procedure

1. Perform the steps in “View final audit schedule” on page 148, 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 “Enter audit data and complete final audit” on page 149.
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.

See also

- “Tools menu in policy file” on page 296

View 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 “View final audit schedule” on page 148.

View 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 “View audit activities” on page 148.

See also

- for more information about the **AuditTask** batch process.
- “Create activities for overdue premium reports” on page 151 for more information about the **Overdue Premium Report** batch process.

Select premium reports

About this task

You can select premium reports on the **Payment** screen of a workers’ compensation submission, rewrite, or renewal policy transaction.

To select premium reports:

Procedure

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**.

Create 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 delinquency. 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.

Enter premium report data

About this task

The insured completes the premium report and returns it to the insurer. The user enters the report data into PolicyCenter.

To enter premium report data:

Procedure

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 following fields. The **Reporting Trend Analysis** also appears on the policy **Summary** screen.

Field	Description
Total Estimated Premium	The pro rata premium based on the number of days reported to date and the Total Estimated Premium for the policy. (Until final audit, there is no total premium on a reporting policy. Until final audit, it is only an estimated premium.)
Total Reported Premium	The total premium from completed premium reports.
Ratio	The ratio between the total estimated premium and total reported premium.
Days Reported	The number of days for which the total reported premium applies.

For more information about reporting trend analysis, see “Premium report trend analysis” on page 143.

Edit or waive a premium report

You can choose to edit or waive a premium report. The steps are the same as for final audit. See “Edit, start, or waive a final audit” on page 148.

Add a new premium report

In certain cases, you can add a new premium report. The section “Adding a premium report” on page 143 describes when you can add a new premium report. The steps are similar to the steps for final audit. See “Add a new audit” on page 149.

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.

See also

- *Configuration Guide*

Side-by-side quoting overview

This includes topics that provide a high-level description of side-by-side quoting.

Side-by-side quoting versus 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 enables 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 in the default configuration.

See also

- “Multi-version quoting” on page 161

Side-by-side availability overview

PolicyCenter supports side-by-side quoting for the following policy transaction types:

- Submission
- Policy change
- Renewal
- Rewrite

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 will display a warning message after you click the **Side-by-Side** button to enter side-by-side mode.

Base data overview in side-by-side quoting

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

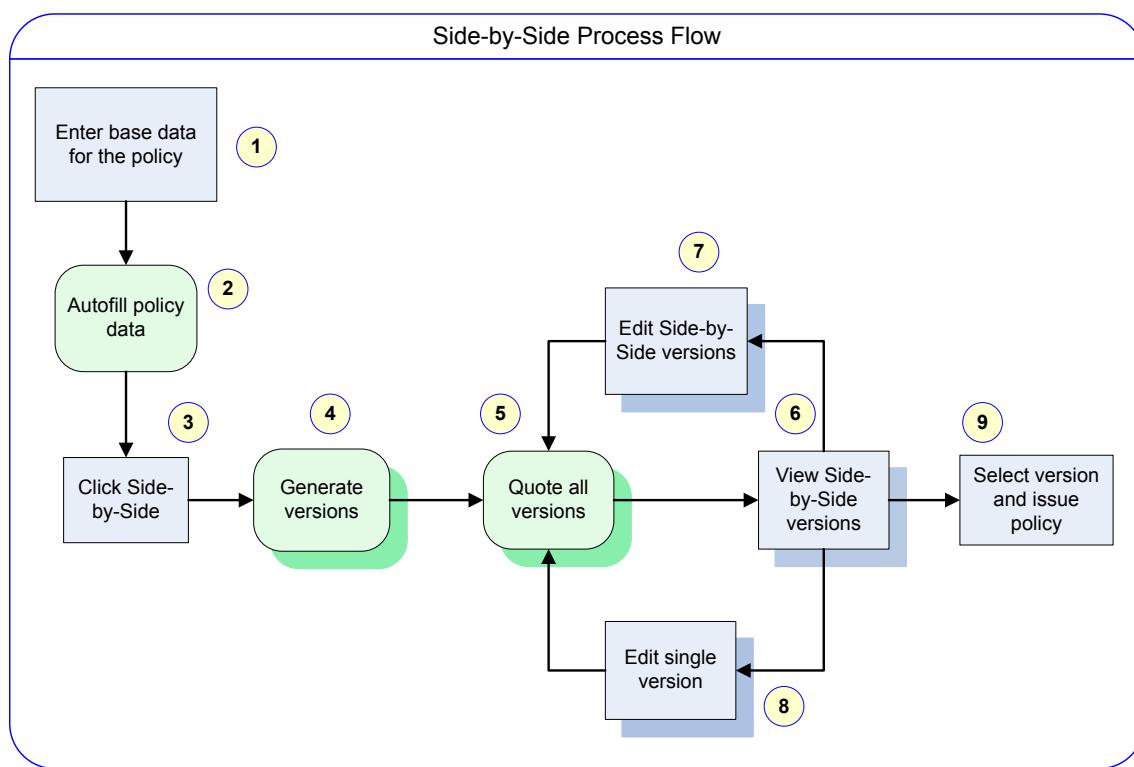
Personal auto side-by-side data

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 selects **Versions**—**Start 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 (“Side-by-side quoting process flow” on page 155).
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 (“Side-by-side quoting process flow” on page 155).
9. The user selects a version and issues the policy.

Side-by-side quoting in the user interface

This includes topics that describe the screens and menu items in the user interface for side-by-side quoting.

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 in which you have selected 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. You can configure this screen to display base data as long as the user can edit it in only one place. Do not repeat base data in the side-by-side columns.

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** (`editlockoverride`) 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 the **Side-by-Side Quoting** 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 the *Configuration Guide*.

- **Rate All** – Generates rates for all versions.
- **Validate All** – Generates validation errors and underwriting issues for all versions.
- **Save All** – Saves all versions.

Quote comparison

Each version has the following fields:

- **Name** – A text box for editing the version name.
- **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 version. You cannot **Rate 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.

Clicking **Select** marks the selected policy period for the policy transaction. Reporting and other PolicyCenter processes may access this status. 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. From a version, select **Versions**→**View Side-by-Side Versions** to return to the **Side-by-Side Quoting** screen.

- **View Period Details** – Jumps to the **Policy Review** screen.
- **Duplicate Period** – Creates a new version based on this version.
- **Delete Period** – Deletes 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 **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 base 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 in side-by-side quoting

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 **Apply All Changes** button handles preemption in side-by-side policy periods. You can view the `HandlePreemptionPopup` in Studio.

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 Name** and click this button.
- **Rename** – To rename a version, select a **Version Name** and click this button.
- **Withdraw** – To withdraw one or more versions, select a **Version Name** 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 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

These topics provide step-by-step instructions for working with side-by-side quoting in PolicyCenter.

Note: The instructions are written for the personal auto line of business.

See also

- “Side-by-Side Quoting screen” on page 156.

Select side-by-side quoting in a submission

About this task

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**.

Procedure

1. Start a submission for personal auto. In the sample data, you can use the Ray Newton account.

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 rated. **Policy Premium** displays the rate.

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 the changed versions.

7. Click **Quote All** to regenerate the **Policy Premium** for this version and to generate rates for all versions.

Edit a version in a policy transaction with side-by-side quoting

About this task

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 Quoting screen” on page 156.

Procedure

1. In a policy transaction with side-by-side quoting, click **Tools**→**Side-by-Side Quoting** to jump to that screen.

2. Under **Version #2**, click **Select** to select this version.

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 **Save Draft**.

9. 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.

10. Under **Version #1**, click **Select** to select this version.
11. In the left sidebar, click **Policy Contract** to go to the **Quick Quote Information** screen for **Version #1**.

Notice that the **Term Type** has the new value that you set for **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.

12. Click **Versions**→**View Side-by-Side Versions**.
13. Click **Rate All** to generate rates for all versions.

Bind and issue a side-by-side submission

About this task

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.

Procedure

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.

Multi-version quoting versus 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. 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 153
- “Side-by-side quoting versus multi-version quoting” on page 153

Working with multi-version quoting

This topic describes how to work with multi-version quoting in the PolicyCenter user interface.

Create and compare multi-version quotes

About this task

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.

Procedure

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. Below the **Actions** menu, drop-down menu displays the name of the current version.

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 message indicating that you are viewing multiple parallel versions and not side-by-side versions. 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. In `config.xml` in Studio, the maximum number of quotes is 3. To change the maximum number of multi-version quotes, change the parameter values. The parameters are:

- `RenewalMaxQuotes`
- `RewriteMaxQuotes`
- `SubmissionMaxQuotes`
- `PolicyChangeMaxQuotes`

See also

- *Configuration Guide*

Part 4

Lines of business

Line of business overview

In the base configuration, PolicyCenter includes several common 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, 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.

In the base configuration, the lines of business are:

Line of business	See also
Commercial auto	"Commercial auto" on page 177
Businessowners	"Businessowners" on page 167
Commercial property	"Commercial property" on page 197
Commercial package policy	"Commercial package policy" on page 189
General liability	"General liability" on page 209
Homeowners	"Homeowners" on page 219
Inland marine	"Inland marine" on page 247
Personal auto	"Personal auto" on page 257
Workers' compensation	"Workers' compensation" on page 273

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 also

- *Product Model Guide*

Affinity groups

Some insurers write policies based on the insured belonging to a certain group. For these insurers, the lines of business in the base configuration enable you to associate an affinity group with a policy. Affinity groups are defined in the **Administration→Users & Security→Affinity Groups** tab. You must have the **Affinity Group Administration** permission. On the **Policy Info** screen, you can select an affinity group from those that meet required criteria (product line, producer code, and date range) and associate it with a policy.

You can associate an affinity group to a policy from among a set of affinity groups that have been defined. In the default configuration, selecting an affinity group does not cause any changes to the policy. Through configuration, you can use affinity groups to compute a dividend, select an offering, or change how premiums are calculated for members of the group.

See also

- “Working with affinity groups” on page 712

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 policy transactions, 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.

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

In 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, coverage terms, coverage term options and packages, modifiers, and question sets.

In the base implementation, the answers to the questions determine the choices on the **Offering Selection** drop-down list.

For more information about offerings, see “Offerings in the product model” on page 450.

Qualification screen for businessowners

The questions reflect risks that the insurer 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
Date Quote Needed	PolicyCenter displays a validation message if this date is in the past.
Estimated Premium	An estimated amount that you can enter prior to quoting. For commercial lines of business in submission, renewal, or rewrite policy transactions. For renewals and rewrites, the value is populated from the value on the prior policy period. PolicyCenter updates the value with the Total Premium when the quote is released. It is also updated when the policy is bound or issued. The value also appears on the Policy Review screen.
Primary Named Insured	PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select: <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact 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. For Existing Contact , PolicyCenter lists account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured .
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	Use to create additional named insureds. 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 This might be a business partner, for example.
Policy Details	Includes fields such as the Term Type , Effective Date , Expiration Date , and Written Date .
Affinity Group	The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search popup. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search popup. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable.

Field name	Description
	<p>The Affinity Groups popup (<code>AffinityGroupSearchPopup.pcf</code>) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the <code>affinitygroupadmin</code> permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab.</p> <p>See <i>Application Guide</i>.</p>
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p>
Producer of Service	<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. In a policy change, you can change the Producer Code.</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. The drop-down list contains a configurable set of choices.</p> <p>For more information about segmentation, see the <i>Configuration Guide</i>.</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>
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, <code>Account.PreferredCoverageCurrency</code>.</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, <code>Account.PreferredSettlementCurrency</code>. You can view and change the preferred settlement currency on the Account Summary screen.</p>

Businessowners line screen

In businessowners, the **Businessowners** screen allows you to enter coverages which apply to the entire policy. This screen has the following tabs:

- **Included Coverages** – Specify coverages that have been defined as **Required** or **Suggested**.
- **Additional Coverages** – Add coverages that have been defined as **Electable**.
- **Exclusions & Conditions** – Add exclusions or conditions.

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 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	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 page of the policy line.

See also

- “Multicurrency features” on page 487

Included Coverages tab

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 Type-lists → 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. PD Deductible – Specify the deductible amount. PD Deductible – Select whether the deductible applies to each claim or each occurrence.
Tenants Fire Liability	PolicyCenter provides a basic limit by default. You can add additional limits to the default application.
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.
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

On 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.

Additional coverages

The **Additional Coverages** tab is similar to the “Businessowners line screen” on page 169. Coverages available on this tab apply only to the selected location.

Location questions

The **Questions** tab contains a question set which you can configure. You can enhance this page to present different question sets based on your business needs.

Buildings screen for businessowners

In businessowners, buildings are the fundamental rating unit for 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** 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.

Details tab

The **Details** tab allows you to enter building attributes that will be used for rating, insurance to value testing, and underwriting information.

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.
Building 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.

Additional Coverages tab

This tab is similar to “Businessowners line screen” on page 169. Coverages on this tab apply only to the selected building.

Modifiers screen for businessowners

In 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

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.
- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.
- **Claims** – 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 784.
- **Prior Losses** – Displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

Policy Review screen for businessowners

For a submission policy transaction, this screen contains all the policy data in summary form. For other policy transactions, this screen displays the differences between the policy versions. In a policy change, out-of-sequence conflicts appear on an **Out-of-Sequence** tab. 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.

The **Policy Review** screen displays coverages, exclusions, policy conditions, and modifiers. Where a value applies, the screen displays the value for each item. If PolicyCenter is configured as a multicurrency system, values are 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 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

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 511.

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 in Rating Management” on page 542.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Forms screen for businessowners

The **Forms** screen displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 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.

See also

- “Working with the Payment screen” on page 775

Businessowners object model

This contains topics that describe the objects or entities associated with the businessowners line of business.

See also

- “Core entities associated with policies” on page 300
- “Cost and transaction model for businessowners line” on page 398

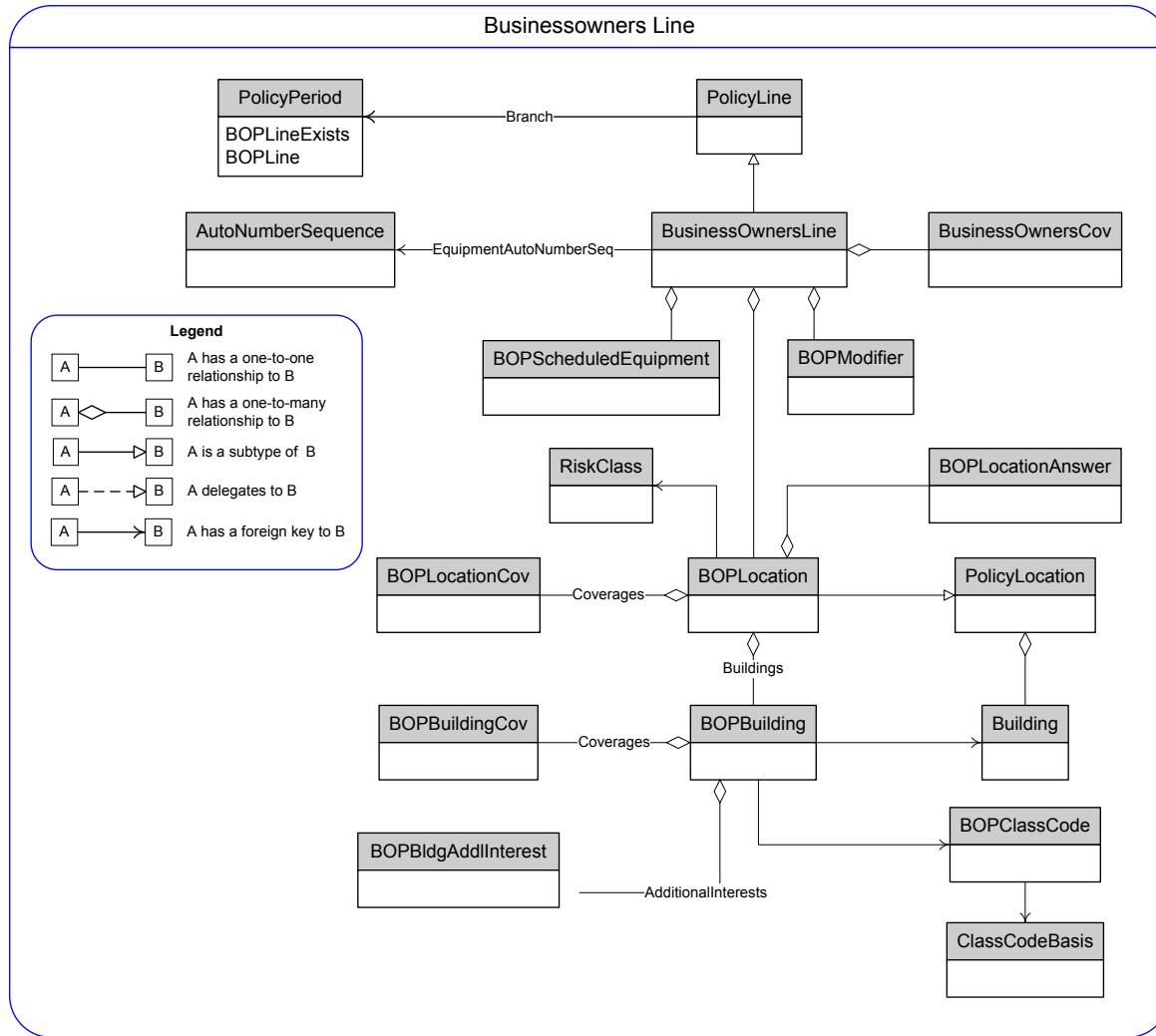
Businessowners object model overview

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 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
BusinessOwnersCov	BusinessOwnersLine	Coverage choices that apply to the entire policy, such as Policywide Property Deductible.
BOPLocationCov	BOPLocation	Coverage choices that apply to a location, such as Money and Securities Cov.
BOPBuildingCov	BOPBuilding	Coverage choices that apply to a building, such as Building Coverage.

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
BOPModifier	BusinessOwnersLine	A modifier of the policy line. In the default application, there is one modifier called Schedule Credit. 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 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 policy transactions, 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.

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.

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

On the **Offerings** screen, you can select an offering. In the base configuration, you can chose between **Standard** and **Special Risk**.

Configuring commercial auto offerings

Use Product Designer to view and change commercial auto offerings.

To view the differences between the standard and special risk offerings, navigate to the **Commercial Auto** product. 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 view the differences between the **Standard** and **Special Risk** offerings, select an offering then select **Selections**. In the **Special Risk** offering, expand the entries under the **Name** column to view how this offering modifies the policy line.

See also

- *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 insurer 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 base 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

In Product Designer, commercial auto has the **Commercial Auto Pre-Qualification** question set.

For information on configuring question sets, adding and removing questions, and specifying correct answers, failure messages, and blocking actions, see 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
Date Quote Needed	PolicyCenter displays a validation message if this date is in the past.
Estimated Premium	An estimated amount that you can enter prior to quoting. For commercial lines of business in submission, renewal, or rewrite policy transactions. For renewals and rewrites, the value is populated from the value on the prior policy period. PolicyCenter updates the value with the Total Premium when the quote is released. It is also updated when the policy is bound or issued. The value also appears on the Policy Review screen.
Primary Named Insur- ed	PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select: <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact 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. For Existing Contact , PolicyCenter lists account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured .
Organization Type	Enables you to select the type of organization such as whether this business is a corporation or partnership.

Field name	Description
Additional Named Insureds	<p>Use to create additional named insureds. 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 <p>This might be a business partner, for example.</p>
Additional Insureds	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.
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>
Affinity Group	<p>The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search popup. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search popup. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable.</p> <p>The Affinity Groups popup (<code>AffinityGroupSearchPopup.pcf</code>) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the <code>affinitygroupadmin</code> permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab.</p> <p>See <i>Application Guide</i>.</p> <p>See “Affinity groups” on page 166.</p>
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p>
Producer of Service	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 . In a policy change, you can change the Producer Code .
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.
	For more information about segmentation, see the <i>Configuration Guide</i> .

Field name	Description
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 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 view and change the preferred settlement currency on the Account Summary screen.

Configuring the Policy Info screen

You can configure the **Policy Info** screen in Studio by navigating to `SubmissionWizard_PolicyInfoDV.pcf`.

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 page of the policy line.

See also

- “Multicurrency features” on page 487

Coverages tab for commercial auto line

On the **Commercial Auto Line** screen, 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**

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 the **Commercial Auto Line** policy 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

On the **Commercial Auto Line**→**Additional Coverages** tab, you can add coverages. You can search for coverages by category. Add a **Keyword** to limit the search to coverages that contains the keyword.

You can configure coverages by navigating in Product Designer to the **Commercial Auto Line** policy 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

On the **Commercial Auto Line**→**Exclusions & Conditions** tab, you can enter line-level exclusions and conditions. In the base 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 the **Commercial Auto Line** policy line. Select **Exclusions** or **Conditions** to add new exclusions or conditions.

Additional Insureds tab for commercial auto line

On the **Commercial Auto Line**→**Additional Insureds** tab, you can enter additional insureds.

Locations screen for commercial auto

Specify the locations for vehicles on the policy.

On 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.

Territory code

In commercial auto policies, locations have a required **Commercial Auto Line Territory Code** field that corresponds to the address of the location. You can configure territory codes in the `territory_codes.xml` system table in Product Designer.

Vehicles screen for commercial auto

In the **Vehicles** screen, you create the 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 for commercial auto

On the **Vehicles** screen in commercial auto, you can enter basic information about the vehicle in the **Vehicle Details** card. Some of the fields on this screen are described below.

Field name	Description
Garaged At	Specify the garage location of the vehicle. The drop-down list contains policy locations and account locations.

Field name	Description
	<p>Click the down arrow button to the right of the list to:</p> <ul style="list-style-type: none"> • Edit the current garage location • Create a new garage location <p>Changing the location updates the location on the account, but does not change locations on other in-force policies.</p>
Vehicle Type	Select a vehicle type. The default configuration includes: <ul style="list-style-type: none"> • Trucks, Tractors, Trailers • Passenger Vehicles • Livery Vehicles • Special
VIN	<p>Enter the vehicle identification number.</p> <p>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.</p> <p>See “PolicyCenter integration points” on page 36 for additional information.</p>
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. The selected vehicle type and garage location affect which modifiers are available.

Configuring the Vehicle 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 the **Commercial Auto Line** policy 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 in commercial auto

On the commercial auto **Vehicles** screen, 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 the **Commercial Auto Line** policy 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 in commercial auto

On the commercial auto **Vehicles** screen, you can 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

Configuring additional coverages

You can configure coverages by navigating in Product Designer to the **Commercial Auto Line** policy line and selecting **Coverages**. To enable a coverage to display on this tab in PolicyCenter, select one of the **Category** types listed above.

State Info screen for commercial auto

For each jurisdiction on the policy that is used as a garaging location, the **State Info** screen has coverages, exclusions, conditions, and jurisdictional rating modifiers.

Coverages tab for states in commercial auto

For each jurisdiction that is a garage location on the policy, the **Coverages** tab on the **State Info** screen has 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 the **Commercial Auto Line** policy 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 in commercial auto

In commercial auto, on the **State Info**→**Additional Coverages** tab, you can add coverages by category. Specify a **Keyword** to limit the search to coverages containing that keyword.

Configuring the additional coverages tab for states

You can configure coverages by navigating in Product Designer to the **Commercial Auto Line** policy 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 in commercial auto

On the **Exclusions & Conditions** tab, you can add exclusions and conditions for each jurisdiction on the policy by category. Enter a **Keyword** to limit the search to exclusions or conditions containing the keyword.

Configuring the Exclusions & Conditions tab for states

You can configure coverages by navigating in Product Designer to the **Commercial Auto Line** policy 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 in commercial auto

On the **State Rating** tab, you can 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 modifiers on the State Rating tab

To configure the state rating modifiers, navigate to the **Commercial Auto Line** policy 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

You can 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.

Covered Vehicles screen for commercial auto

In 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 the **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: **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 the **Commercial Auto Line** policy 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, the modifiers are:

- **Liability Rates**
- **Physical Damage Rates**

Each credit/debit category 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 modifiers on the Modifiers screen

To configure the rating modifiers, navigate to the **Commercial Auto Line** policy 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

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.
- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.
- **Claims** – 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 784.
- **Prior Losses** – 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

In 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 policy transaction, this screen contains all the policy data in summary form. For other policy transactions, this screen displays the differences between the policy versions. In a policy change, out-of-sequence conflicts appear on an **Out-of-Sequence** tab. 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.

The **Policy Review** screen displays coverages, exclusions, policy conditions, and modifiers. Where a value applies, the screen displays the value for each item. If PolicyCenter is configured as a multicurrency system, values are in the currency set on the coverable.

Quote screen for commercial auto

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen. This screen 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

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 511.

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 in Rating Management” on page 542.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Forms screen for commercial auto

The **Forms** screen displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 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.

See also

- “Working with the Payment screen” on page 775

Commercial auto object model

This includes topics that describe the objects or entities associated with the commercial auto line of business.

See also

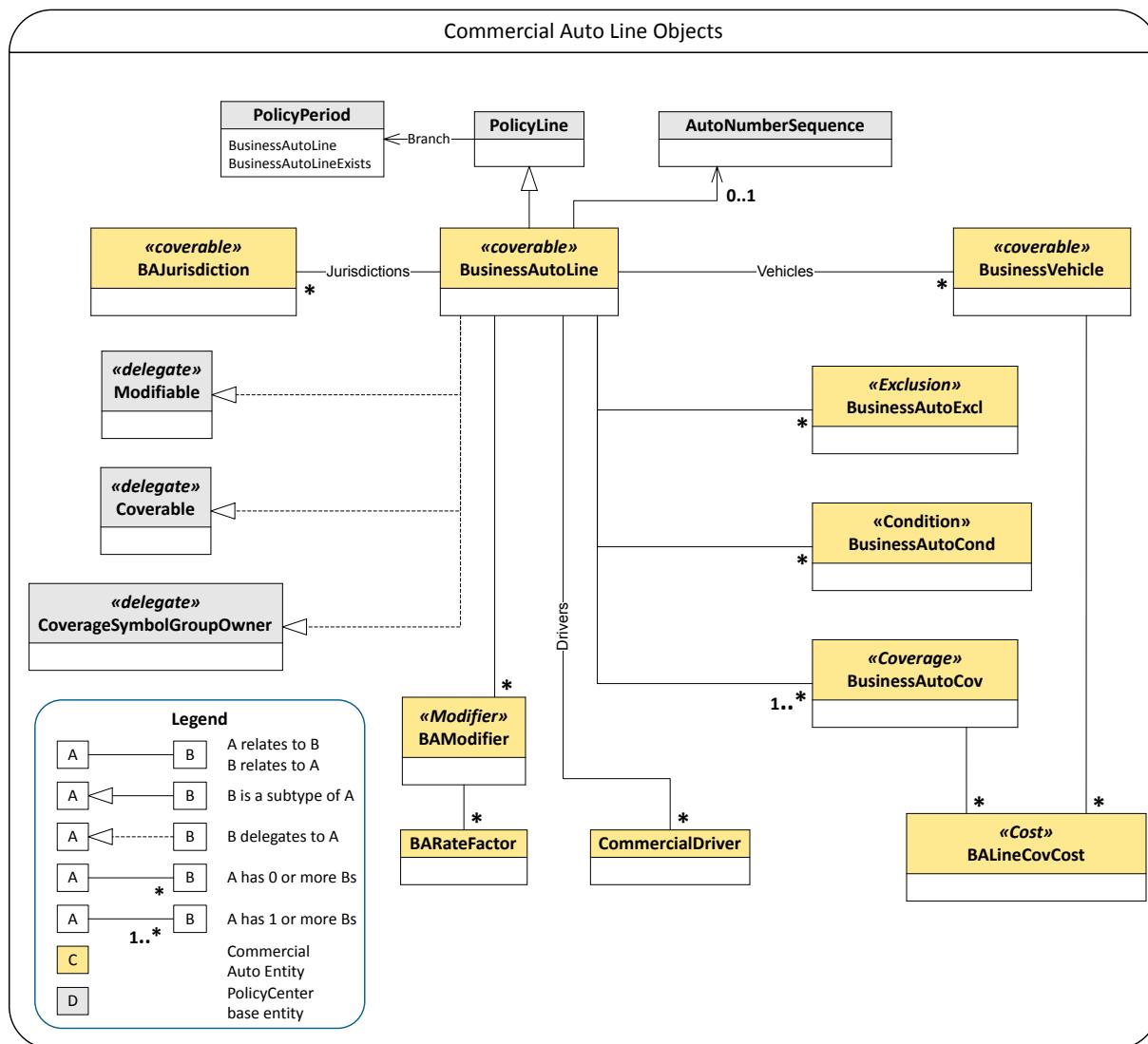
- “Core entities associated with policies” on page 300
- “Cost and transaction model for commercial auto line” on page 400

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
BusinessAutoCov	BusinessAutoLine	Coverage choices that apply to the entire policy.
BASStateCov	BAJurisdiction	Coverages that apply to jurisdictions. This coverage has one subtype: <ul style="list-style-type: none"> • BAHiredSpecPerilCov

Coverage type	Attaches to	Description
BusinessVehicleCov	BusinessVehicle	Coverages for vehicles. This coverage has one subtype: <ul style="list-style-type: none">• BASpecCausesLossCov

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.

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 policy transactions, 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.

See also

- “Commercial property” on page 197
- “General liability” on page 209
- “Inland marine” on page 247

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 includes topics that describe the screens in commercial package policy.

Offerings screen for commercial package

In 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, coverage terms, coverage term options and packages, modifiers, and question sets.

In the base implementation, the answers to the questions determine the choices on the **Offering Selection** drop-down list.

For more information about offerings, see “Offerings in the product model” on page 450.

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:

- Premium Program
- Special Risk
- Standard

Selecting **Special Risk** removes inland marine from the policy.

Qualification screen for commercial package

The **Qualification** screen contains a set of questions to pre-qualify the applicant.

The questions reflect risks that the insurer 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 base 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
Date Quote Needed	PolicyCenter displays a validation message if this date is in the past.
Estimated Premium	An estimated amount that you can enter prior to quoting. For commercial lines of business in submission, renewal, or rewrite policy transactions. For renewals and rewrites, the value is populated from the value on the prior policy period. PolicyCenter updates the value with the Total Premium when the quote is released. It is also updated when the policy is bound or issued. The value also appears on the Policy Review screen.
Primary Named Insured	PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select: <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact 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. For Existing Contact , PolicyCenter lists account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured .

Name of field	Description
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. 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 <p>This might be a business partner, for example.</p>
Policy Details	Includes fields such as the Term Type , Effective Date , Expiration Date , and Written Date .
Affinity Group	<p>The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search popup. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search popup. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable.</p> <p>The Affinity Groups popup (<code>AffinityGroupSearchPopup.pcf</code>) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the <code>affinitygroupadmin</code> permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab.</p> <p>See <i>Application Guide</i>.</p>
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p>
Producer of Service	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 . In a policy change, you can change the Producer Code .
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.
	For more information about segmentation, see the <i>Configuration Guide</i> .
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 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, <code>Account.PreferredCoverageCurrency</code>.</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

Name of field	Description
	The currency choices are populated from the Currency typelist. The default is the preferred settlement currency on the account, Account.PreferredSettlementCurrency. You can view and change the preferred settlement currency on the Account Summary screen.

Line Selection screen for commercial package

In commercial package, on the **Line Selection** screen, you can enable and disable lines, and 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 policy line in commercial package policy based on the selected offering. The **Special Risk** offering removes the inland marine line.

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

On 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 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 **Territory Code for General Liability Line** field. If the commercial property line is enabled, locations have an additional **Territory Code Commercial Property Line** field.

Line of business screens for commercial package

In commercial package, the left sidebar displays each line of business that you selected in the **Line Selection** screen. You can click the name of a line of business to reveal the 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. 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.

See also

- “General liability screens” on page 211
- “Commercial property screens” on page 199
- “Inland marine screens” on page 248

Modifiers screen for commercial package

In commercial package, the **Modifiers** screen displays modifiers defined at the product level. (General liability and commercial property have their own **Modifiers** screens for modifiers that apply to the line.) The commercial package product has one modifier for an individual risk premium modification (**Package IRPM**). The rate factors for this modifier allow you to adjust the rating based on:

- **Building features**
- **Employees**
- **Location**
- **Management**
- **Premises and equipment**
- **Protection**

Risk Analysis screen for commercial package

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.
- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.
- **Claims** – 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 784.
- **Prior Losses** – 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 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

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 511.

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 in Rating Management” on page 542.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Policy Premium tab on Quote screen for commercial package

On the commercial package **Quote** screen, the **Policy Premium** tab displays the premium for each line of business in the policy.

You can use the **Override Rating** button to override the premium that the rating engine automatically generates for a policy. This option is only available for inland marine, because it is the only line in commercial package that has rating overrides in the base configuration. For more information, see “Rating overrides” on page 411.

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 `RatingPanel1Set.CommercialPackage.ILMLine.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 displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 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.

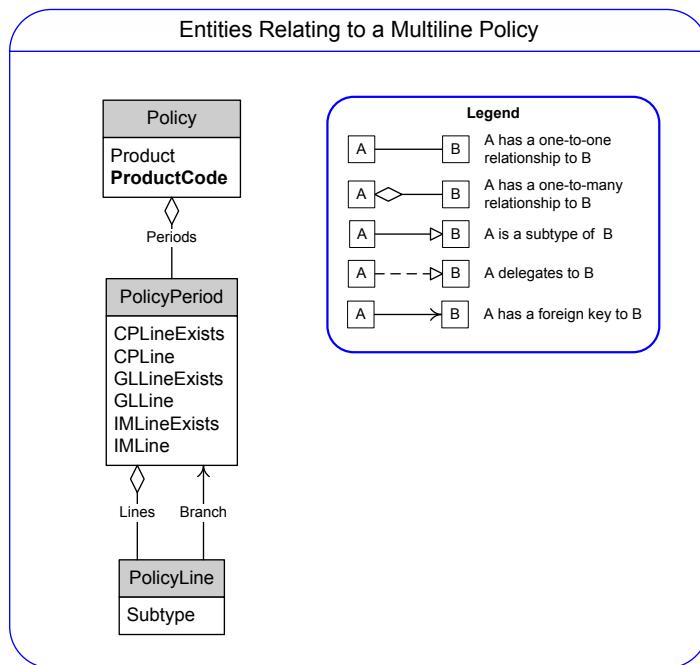
See also

- “Working with the Payment screen” on page 775

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**. In Product Designer, the **ProductCode** in the **Code** on the **Product Model→Products** page.

The **PolicyPeriod** entity has an *LNLinExists*, where *LN* is an abbreviation for the line. This derived property returns true if a line exists on the policy period. The *LNLin* derived property allows you to retrieve the policy line. The **PolicyPeriod** entity also has a **Lines** array to each **PolicyLine**.

See also

- “Working with policies” on page 298
- “Commercial property object model” on page 204
- “General liability object model” on page 217
- “Inland marine object model” on page 253

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 Product Designer, the following items in the **Product Model** are related to commercial package policy:

Page	Item
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** page, **Offering Required** is set to true. This setting means that you must select an offering on the **Offerings** screen in PolicyCenter. In Product Designer, you define commercial package offerings on the **Offerings** tab.

In Product Designer, you select the policy lines for this product on the **Policy Lines** page. The policy lines in this product are **Commercial Property Line**, **General Liability Line**, and **Inland Marine Line**.

In Product Designer, you include questions sets for each line on the **Question Sets** page. Also included are offering questions that are related to the commercial package product and are not specific to any particular policy line.

In Product Designer, you define product level modifiers on the **Modifiers** page. 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 435.

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, insurers 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 policy transactions, 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.

Working with commercial property

This topic provides step-by-step instructions on how to work with a commercial property policy in the user interface.

Add locations and buildings in commercial property

About this task

You can add new or existing buildings and locations in a commercial property policy transaction. You must specify at least one building for each location.

Procedure

1. In a commercial property policy, navigate to the **Buildings and Locations** screen.

Add location

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.

Add building

3. 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.

Copying coverages to other buildings in commercial property

About this task

In a commercial property policy transaction, 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.

Procedure

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 coverages for commercial property” on page 208

Add blanket coverages in commercial property

Procedure

1. Navigate to the **Blankets** screen and click **Add Blanket** to display the **Blanket Details** screen.
PolicyCenter provides an autonumber for the blanket.
2. Enter an optional **Description**.
3. Use the **Blanket Type** drop-down menu to apply blanket coverage. Choices are:
 - **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.
4. Use the **Group Type** drop-down menu select the blanket type. Choices are:
 - **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 define the **Group Type** choices in the product model in Studio.

5. Specify **Limit**, **Deductible**, and **Coinsurance** percentages.
6. Click **Show Coverages** to display the coverages in the blanket.
7. 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.

Configure blankets for commercial property in the product model

About this task

In Product Designer, you can configure blankets in the **Commercial Property Line** policy line. The **CPBlanket Coverage** coverage provides an example of a blanket.

Procedure

1. Navigate to a coverage such as **CPBlanket Coverage**.
2. Select **Terms** to view the limit, deductible, and coinsurance.
3. 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 in PolicyCenter.

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 includes topics that describe the screens in commercial property.

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
Date Quote Needed	PolicyCenter displays a validation message if this date is in the past.
Estimated Premium	An estimated amount that you can enter prior to quoting. For commercial lines of business in submission, renewal, or rewrite policy transactions. For renewals and rewrites, the value is populated from the value on the prior policy period. PolicyCenter updates the value with the Total Premium when the quote is released. It is also updated when the policy is bound or issued. The value also appears on the Policy Review 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 account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured.</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. 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 <p>This might be a business partner, for example.</p>
Policy Details	Includes fields such as the Term Type , Effective Date , Expiration Date , and Written Date .
Affinity Group	<p>The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search popup. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search popup. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable.</p> <p>The Affinity Groups popup (AffinityGroupSearchPopup.pcf) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the affinitygroupadmin permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab.</p> <p>See <i>Application Guide</i>.</p>
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p>
Producer of Service	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 . In a policy change, you can change the Producer Code .
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.
	For more information about segmentation, see the <i>Configuration Guide</i> .
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 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>

Name of field	Description
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 view and change the preferred settlement currency on the Account Summary screen.</p>

Buildings and Locations screen for commercial property

In 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.)

The buttons at the top of the screen are:

- **Add Location** – See “Add locations and buildings in commercial property” on page 197.
- **Remove Buildings**
- **Copy Coverages** – See “Copying coverages to other buildings in commercial property” on page 198.
- **Spreadsheet** – Import or export buildings and locations from a policy in PolicyCenter to a spreadsheet in .xlsx format. See “Using spreadsheets generated by policy data spreadsheet import/export” on page 439.

For each building, the **Building and Locations** screen displays the limit for the following coverages:

- **Building Limit** column displays **Building Coverage**
- **Business Personal Prop Limit** column displays **Business and Personal Property Coverage**
- **Business Income Limit** column displays **Business Income Coverage**
- **Extra Expense Limit** column displays **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 page of the policy line.

See also

- “Multicurrency features” on page 487

Location Information screen for commercial property

In commercial property, the **Location Information** screen has information about a location. In commercial property, the base configuration has an integration with Guidewire Spotlight interactive and risk assessment services.

See also

- “Locations” on page 327
- “Risk assessment with Spotlight” on page 431

Building screen for commercial property

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.

On the **Coverage Form** drop-down menu, you can choose from the following choices:

- **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)**

On the **Coverages** tab, you can add or remove coverages, and specify coverage details such as the limit.

On the **Additional Interest** tab, you can 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.

See also

- “Add blanket coverages in commercial property” on page 198
- “Configure blankets for commercial property in the product model” on page 199

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

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.

- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.
- **Claims** – 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 784.
- **Prior Losses** – Displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

The base configuration does not define 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 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 policy transaction, this screen contains all the policy data in summary form. For other policy transactions, this screen displays the differences between the policy versions. In a policy change, out-of-sequence conflicts appear on an **Out-of-Sequence** tab. 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.

The **Policy Review** screen displays coverages, exclusions, policy conditions, and modifiers. Where a value applies, the screen displays the value for each item. If PolicyCenter is configured as a multicurrency system, values are 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 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

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 511.

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 in Rating Management” on page 542.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Policy Premium tab on Quote screen in commercial property

On the commercial property **Quote** screen, 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 411.

Rating in commercial property

In the default configuration of commercial property, 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

Forms screen for commercial property

The **Forms** screen displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 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.

See also

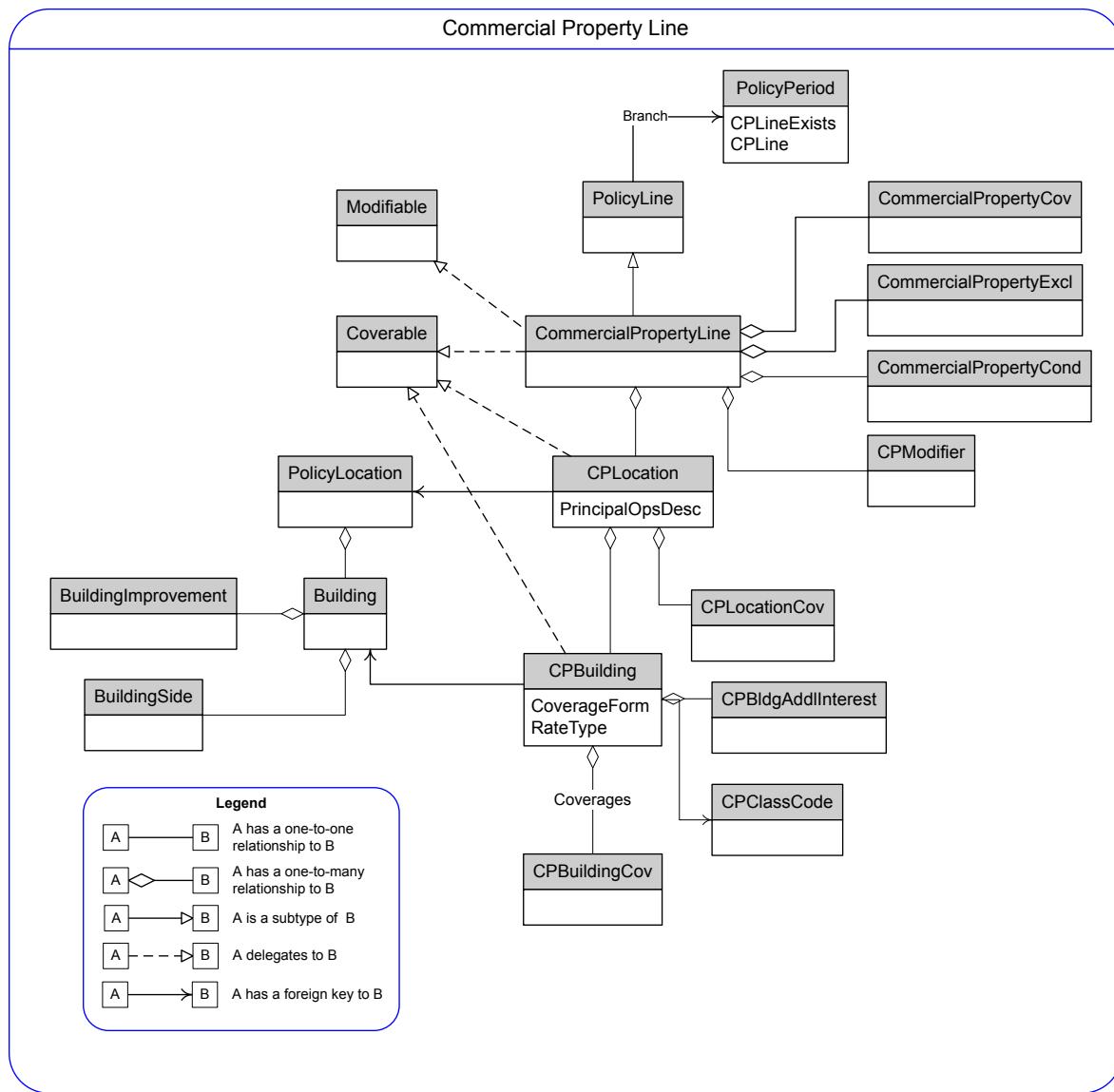
- “Working with the Payment screen” on page 775

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

- “Working with policies” on page 298
- “Cost and transaction model for commercial property line” on page 400

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 206.

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 207.

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.
Business Personal Property Coverage	This coverage provides insurance for the following types of property that are located in the building or close to the building: <ul style="list-style-type: none"> • Stock • Machinery & Equipment • Fixtures, improvements and alterations • Tenants Betterments and Improvements • Property of Others This is a suggested coverage that is not available if the coverage form is condominium unit-owners.
Business Personal Property - Separation of Coverage (Stock) Coverage	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. This is a suggested coverage that is not available if the coverage form is condominium unit-owners or condominium association.
Extra Expense Coverage	This coverage provides insurance against extra expenses. It is a suggested coverage.

Modifier entity in commercial property

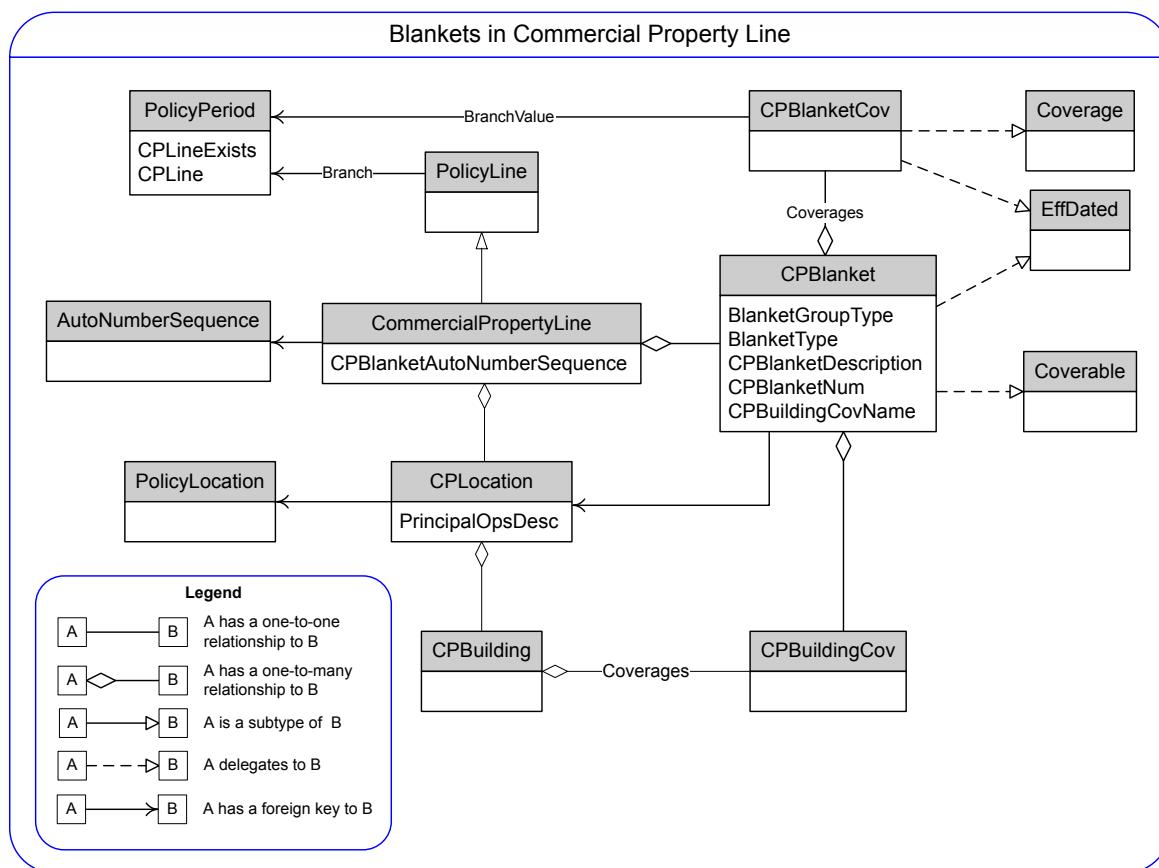
The **CPModifier** entity represents modifiers on the commercial property policy line. In the default configuration, the commercial property line has a single modifier defined, **CPScheduleCredits**.

You can view and configure this modifier in Product Designer in the **Commercial Property Line** policy 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 copying coverages for commercial property

In commercial property, when you 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 policy transactions, 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.

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.

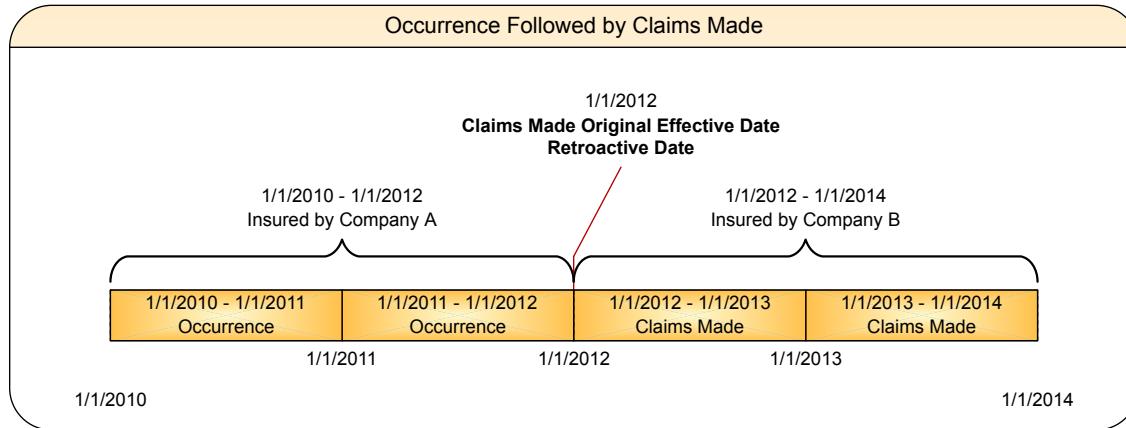
As is typical of a liability policy, all coverages are line-level coverages. At a minimum, each general liability policy contains basic general liability coverages. In addition, the policy can contain additional coverages, exclusions, conditions, and additional insureds. Additional coverages provide 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 policy basis examples

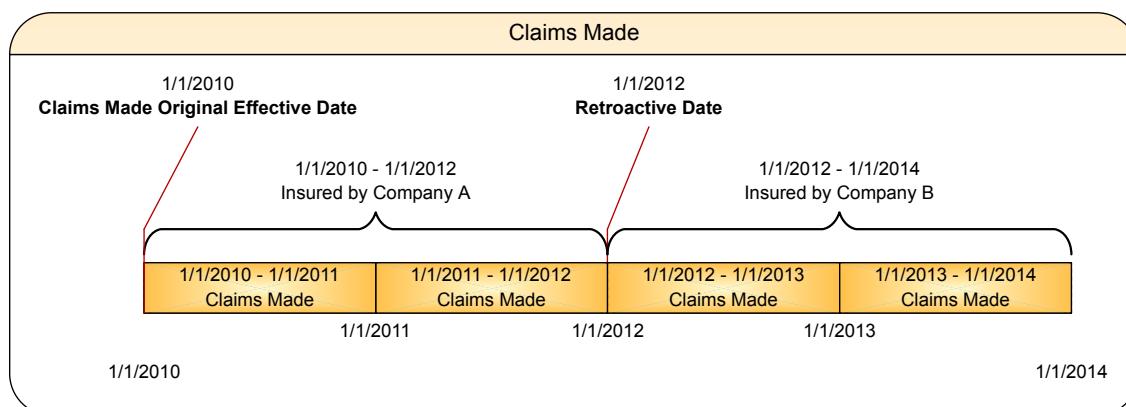
In general liability, you specify policy basis on the **Coverages→Standard Coverages** tab.

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 insurers. The **Claims Made Original Effective** date is set 1/1/2010, the first date that the insured had a claims made policy (with any insurer). The **Retroactive Date** is set to 1/1/2012, the first date the insured has a claims made policy with Company B.

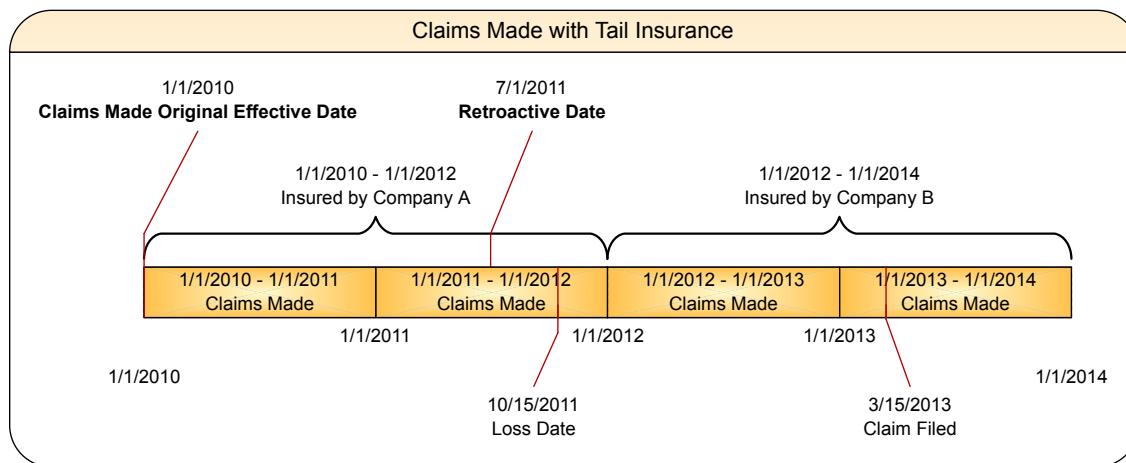
If the **Retroactive Date** is later than the **Claims Made Original Effective Date**, then there is a gap in coverage. For example, a loss occurred after the **Claims Made Original Effective Date** but before the **Retroactive Date**. The claim is filed after the retroactive date. Company A does not 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.



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 includes topics that describe the screens in general liability.

Qualification screen for general liability

In general liability, this screen contains a set of questions to pre-qualify the applicant.

The **Qualification** screen contains a set of questions to pre-qualify the applicant.

The questions reflect risks that the insurer 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 base 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
Date Quote Needed	PolicyCenter displays a validation message if this date is in the past.

Name of field	Description
Estimated Premium	An estimated amount that you can enter prior to quoting. For commercial lines of business in submission, renewal, or rewrite policy transactions. For renewals and rewrites, the value is populated from the value on the prior policy period. PolicyCenter updates the value with the Total Premium when the quote is released. It is also updated when the policy is bound or issued. The value also appears on the Policy Review screen.
Primary Named Insured	PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select: <ul style="list-style-type: none">• New Company• New Person• From Address Book• Existing Contact 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. For Existing Contact , PolicyCenter lists account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured .
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. 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 This might be a business partner, for example.
Policy Details	Includes fields such as the Term Type , Effective Date , Expiration Date , and Written Date .
Affinity Group	The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search popup. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search popup. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable. The Affinity Groups popup (AffinityGroupSearchPopup.pcf) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the affinitygroupadmin permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab. See <i>Application Guide</i> .
Producer of Record	The Organization defaults to the producer that you selected on the New Submissions screen. You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change. Although you can change the values for Organization and Producer Code , PolicyCenter limits your choices by user permissions.
Producer of Service	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 . In a policy change, you can change the Producer Code .
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. For more information about segmentation, see the <i>Configuration Guide</i> .

Name of field	Description
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 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 view and change the preferred settlement currency on the Account Summary screen.

Locations screen for general liability

On 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.

Territory code

In general liability policies, locations have an additional **Territory Code for General Liability Line** field that contains the territory code for the location. The territory code can have an associated rating factor.

Coverages screen for general liability

In general liability, the **Coverages** screen allows you to add standard and additional liability coverages as well as exclusions, conditions, and additional insureds.

Standard Coverages tab

In general liability, the **Coverages**→**Standard Coverages** tab displays coverages that are used by most policies.

Field	Description
Policy	Select either Claims Made or Occurrence .
Basis	<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. 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.) 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 insurer provides coverage for claims which occurred in the past. • 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.

Field	Description
	<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>

Additional Insureds tab

In general liability, the **Coverages**→**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.

Configure standard and additional coverages in general liability

About this task

In PolicyCenter, the **Coverages**→**Standard Coverages** tab displays the **General Liability**, **GL Deductible**, and **Arbitration** coverages. You can configure which coverages appear on the **Standard Coverages** tab.

In PolicyCenter, the **Coverages**→**Additional Coverages** tab allows you to select coverages in addition to the coverages on the **Standard Coverages** tab.

Procedure

1. In Product Designer, open the **General Liability Line** policy line.
2. Go to **Coverages** and click to view a coverage.

Standard coverage

3. To define a coverage as standard, set **Category** to **GL REQUIRED**.

Additional coverage

4. To define a coverage as additional, set **Category** to a value other than **GL REQUIRED** in Product Designer.

Exposures screen for general liability

In general liability, exposures provide the basis for rating. On the **Exposures** screen, 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 a general liability exposure in a policy change

In a general liability 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 the `gl_class_code.xml` system table in Product Designer.

Basis types are specified in **ClassCodeBasis**. You can view the basis types in the `class_code_basis.xml` system table in Product Designer. If the **AuditTable** field is **true**, then exposures using class codes with that basis type are basis scalable, and can be split or ended.

Split exposure in general liability policy change

About this task

The **End** and **Split** buttons appear on the **Exposures** screen in a policy change.

Procedure

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

In 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

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.

- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.
- **Claims** – 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 784.
- **Prior Losses** – 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

For a submission policy transaction, this screen contains all the policy data in summary form. For other policy transactions, this screen displays the differences between the policy versions. In a policy change, out-of-sequence conflicts appear on an **Out-of-Sequence** tab. 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.

The **Policy Review** screen displays coverages, exclusions, policy conditions, and modifiers. Where a value applies, the screen displays the value for each item. If PolicyCenter is configured as a multicurrency system, values are in the currency set on the coverable.

Locations have **Exposure Type** and **Exposure Value** columns. The **Exposure Type** displays the *basis units* for the coverable. Basis units are units of risk. The **Exposure Value** displays the number of basis units.

See also

- “Multicurrency and basis units” on page 492

Quote screen for general liability

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen. This screen 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

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 511.

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 in Rating Management” on page 542.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Forms screen for general liability

The **Forms** screen displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 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.

See also

- “Working with the Payment screen” on page 775

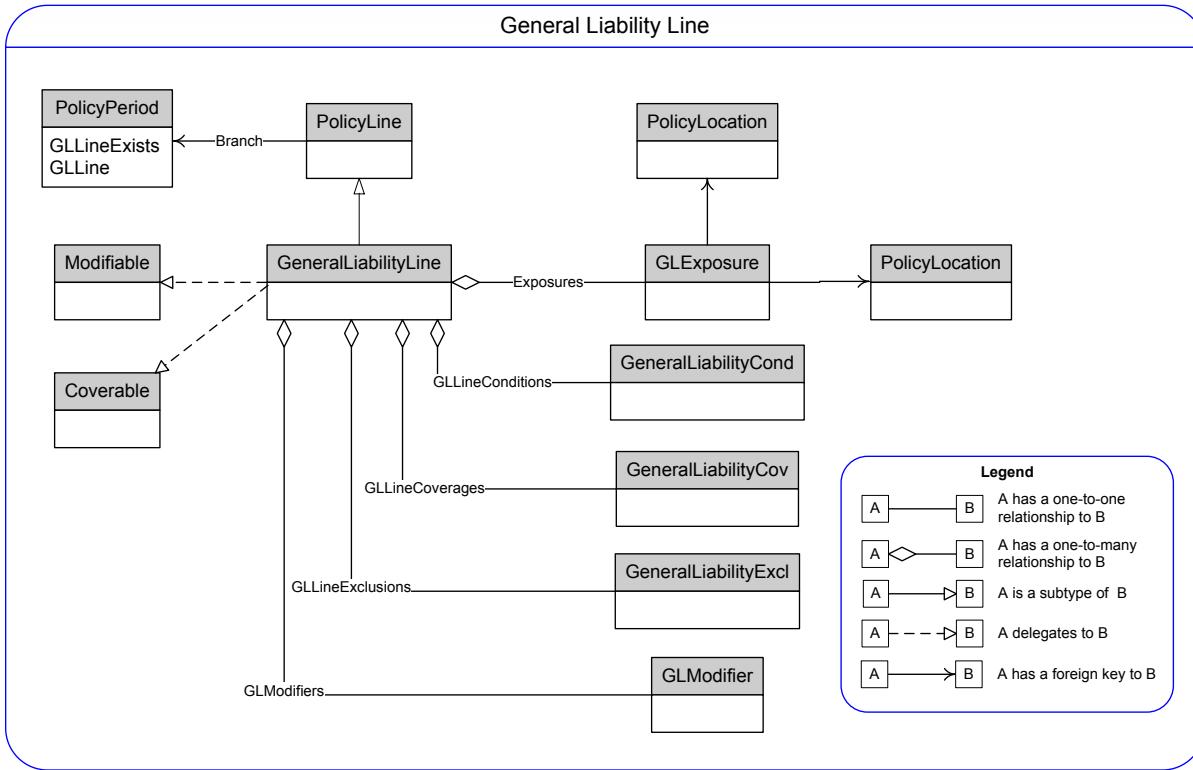
Audits in general liability

In the general liability **Payments→Audits**, 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 300
- “Cost and transaction model for general liability line” on page 401

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 the **General Liability Line** policy line.

Modifier entity in general liability

The **GLModifier** entity represents modifiers on the general liability policy line. You can configure modifiers for general liability in Product Designer on the **Modifiers** page of the product or policy line.

Homeowners

Homeowners insurance provides protection against the financial consequences of losses related to owning and renting a home. A homeowners policy is a combination of property and liability coverages.

Homeowners overview

In the base implementation, the homeowners line is designed according to the United States market. You can use the base implementation as a starting point for implementing homeowners in the United States and for other countries. For example, homeowners in the United States sets certain default limits as percentages of a primary limit. You can remove this dependency and these default limits.

The homeowners implementation provides policy types for insuring dwellings, rentals, and condominiums. Policy types are implemented as coverage parts.

Coverage form

In addition to policy type, homeowners provides *coverage forms*. Coverage forms further the patterns available on the policy. Coverage form availability is related to policy type.

Note: Within the industry the term *form* is used to describe a category offered on a product. However, within PolicyCenter, the term form is used in a different way to describe the physical contracts that define coverages and endorsements. To avoid confusion, Guidewire uses the term *coverage form* to describe the categories offered on a product that in the industry is typically referred to as a *form*.

There are a number of homeowners coverage forms in common use across the United States. In the base configuration, homeowners includes the following coverage forms:

- HO2 – Broad Form
- HO3 – Special Form, most commonly issued
- HO4 – Tenant Contents, also known as Renters
- HO5 – Comprehensive
- HO6 – Condominium Unit Owners

The following table shows the coverage forms available for each policy type:

Policy type	Coverage form
Dwelling	HO2, HO3, HO5
Rental	HO4

Policy type	Coverage form
Condominium	HO6

States

In the base configuration, homeowners provides a generic version of each policy type that applies to all jurisdictions. Some United States state-specific examples are provided. For example, homeowners includes some deductible coverage terms and coverage term options that vary by state. However, complete state data is not included. Through configuration, you can create implementations for specific jurisdictions.

Homeowners dwellings per policy

In the user interface, homeowners allows you to specify only one dwelling per policy. This approach simplifies the user workflow and is in line with standard homeowners policies in the United States, which also allow only one dwelling to be insured per policy.

However, you can extend the implementation to accept multiple dwellings per policy if needed. The underlying data model supports multiple dwelling policies. You can configure the user interface and extend the Gosu code to support multiple dwellings.

Homeowners jurisdictions

In the base configuration of homeowners, there is only one jurisdiction (U.S. state) per policy.

If you extend homeowners to cover multiple dwellings in a single policy, the underlying data model and Gosu code supports dwellings in more than one jurisdiction.

Schedule endorsements for homeowners

In homeowners, several endorsements on homeowners policies require a schedule of objects to be included. Examples are schedules of personal property either on or off the premises, and schedules of other structures either on or off the premises.

Rating

Homeowners includes schedules that require various types of rating. The base configuration includes examples of each schedule type to illustrate how to handle the different types of schedule endorsements.

In some schedules, rating applies the premium at the schedule level, covering all items in the schedule. In other cases, such as for Scheduled Personal Property, rating applies premium at the individual item level. And in some cases, such as jewelry or fine arts, rating applies premium at the item group or type level.

Forms scope for homeowners

In homeowners, the base configuration provides a small sample of forms that illustrate different types of forms inference logic. For more information, see “Inference tab for form patterns” on page 753.

Coverages, exclusions, and conditions for homeowners

In homeowners, the base configuration includes coverages, exclusions, and conditions for the United States market. Homeowners uses these categories to filter searches and to reduce the potential number of items in lists.

Authority profiles and underwriting rules for homeowners

The base configuration includes examples of authority profiles and underwriting rules. In homeowners, each insurer may define authority profiles and underwriting rules relating to risk selection and underwriting behavior.

Refer to “Underwriting rules” on page 643 for information on how to configure these rules to meet your specific needs.

Full application quote for homeowners

In homeowners, the base configuration accepts only full application quote types. No separate Quick Quote configuration is provided.

Homeowners rating

Homeowners includes two demonstration rating implementations:

- A Guidewire Rating Management implementation that provides a basic structure as well as examples for further development of rating. However, this is not a complete implementation of homeowners rating.
- A minimal Gosu-based implementation which is not suitable as a basis for further development. You can use this implementation for demonstration purposes. Prior to integrating with an external rating engine, this implementation enables you to bind and issue homeowners policies while working on other parts of PolicyCenter.

Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter.

Guidewire Rating Management for homeowners

Guidewire Rating Management includes an implementation for homeowners. This partial implementation is similar to homeowners rating specified by ISO.

Disclosing base premium amounts is a particular requirement of homeowners insurance. The base premium value is calculated on the base coverages for the policy, including the default limits and terms for those coverages. When non-default limits or terms are selected for base coverages, these adjustments to the base premium are listed separately.

Rating first calculates and displays a base premium using default values for required section 1 and section 2 coverages. You specify default values for these coverages, and other product model patterns, in Product Designer.

Next, rating calculates adjustments to the base premium for non-default values for section 1 and section 2 coverages. Then rating calculates the remainder of the premium (everything not included in section 1 and section 2 coverages).

See also

- “Policy Premium tab on Quote screen” on page 227

Homeowners offerings

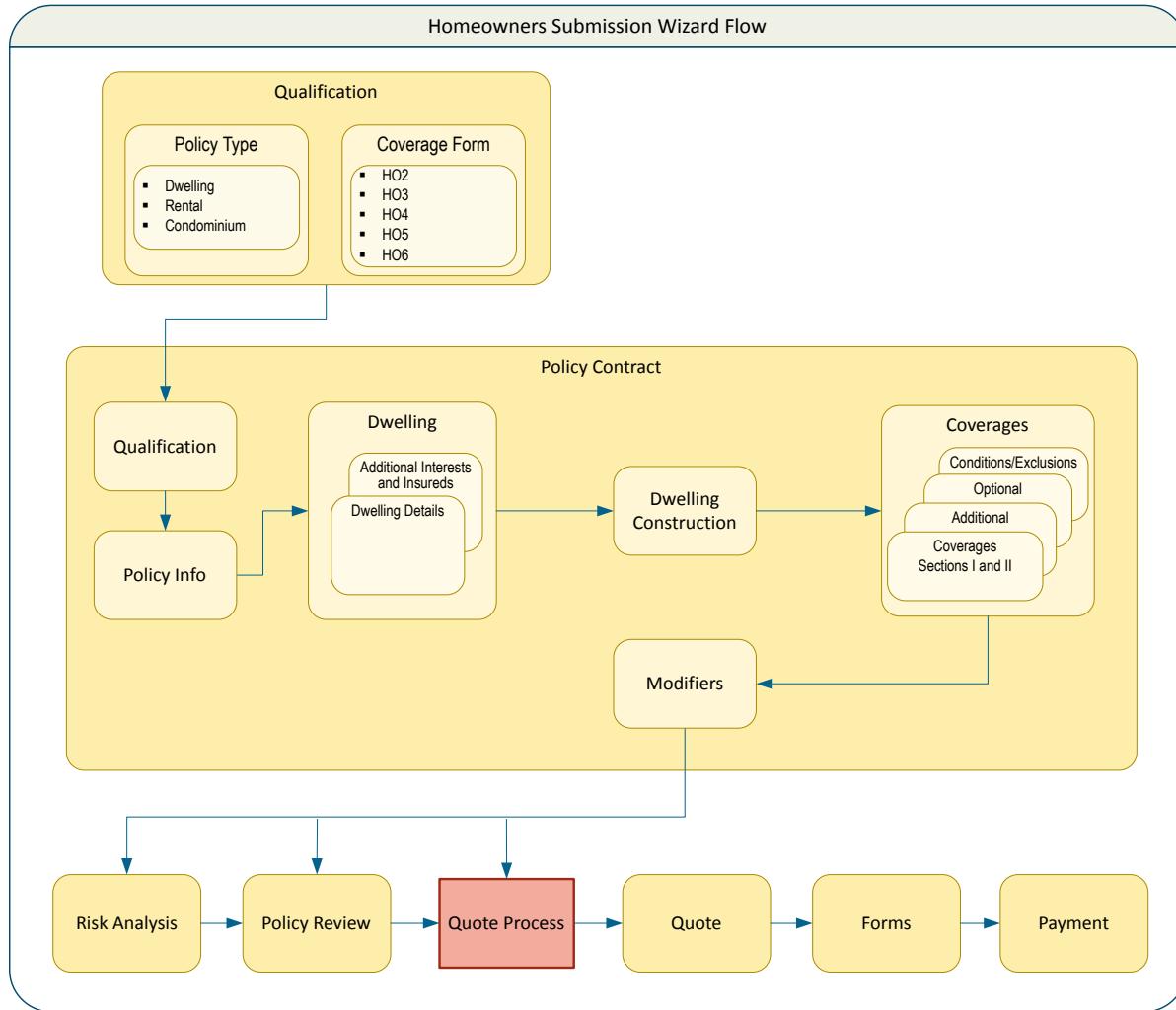
The base configuration of homeowners does not include offerings. Through configuration, you can implement offerings to meet your specific requirements.

Homeowners integrations

The base configuration of homeowners does not include any integrations. However, the homeowners workflow is designed to anticipate integrations such as Interactive Voice Response (IVR), territory class derivation, and protection class derivation. Other standard PolicyCenter integrations, such as address book, rating engine, print/issuance system, and so forth, continue to apply.

Homeowners wizard flow

The following illustration shows the path through the homeowners submission wizard that occurs by clicking **Next** on each screen. Subject to validation, you can navigate directly to any page in the flow by clicking the links in the left sidebar.



There is nothing unique about the wizard flow for other job types.

Homeowners screens

The homeowners line implementation contains a series of screens which enable 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.

Note: The small sample data set contains a homeowners policy for Alicia Shirley. Alice Applegate (aapplegate) is the user assigned to this policy.

Qualification screen for homeowners

This screen contains a **Policy Type** selection list and a set of questions to pre-qualify the applicant. The policy type you select determines the choices available on the **Coverage Form** selection list. The coverage form selection determines coverage and coverage term availability and potentially other things. The selected policy type and coverage form therefore determines the content of many of the screens in the homeowners line.

Note: **Policy Type** is editable only during a submission policy transaction. It is not appropriate to change during other types of policy transactions, because doing so requires a new policy.

The questions are samples that reflect the risks the insurer wants to assess at the beginning of the submission. Since this is the gateway to the remainder of the submission, the intent is to determine eligibility of the applicant and the applicant's property. Answers to all questions are required. Undesired answers can block the submission. You can

change these actions in Product Designer. Some answers must correspond with data entered elsewhere in the Submission. Inconsistent answers result in validation warnings, but not errors. None of the answers add risk points or raise underwriting issues.

In the default implementation, the choice questions use filter statements to enable you to enter details if answered **Yes**.

Configuring the homeowners Qualification screen

The questions in the homeowners **Qualification** screen are defined in the **HOP Pre-Qualification** question set which defines questions that apply to all policy types.

Using Product Designer, you can add, edit, and delete questions and change question behavior as needed. You also can apply filters to control the conditions, add supplemental text, and set availability based on date ranges, jurisdiction, or job type.

Policy Info screen for homeowners

The **Policy Info** screen captures basic information about the policy.

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
Date Quote Needed	PolicyCenter displays a validation message if this date is in the past.
Primary Named Insured	PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select: <ul style="list-style-type: none">• New Company• New Person• From Address Book• Existing Contact 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. For Existing Contact , PolicyCenter lists account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured .
Secondary Named Insured	Use to create the secondary named insured. The drop-down menu enables you to select: <ul style="list-style-type: none">• New Person• From Address Book• Existing Contact Secondary named insured might be a spouse or a child, for example.

Field name	Description
Additional Named Insureds	<p>Use to create additional named insureds. 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 <p>This might be a business partner, for example.</p>
Policy Details	<p>Includes fields such as the Term Type, Effective Date, Expiration Date, and Written Date.</p>
Affinity Group	<p>The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search pop-up. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search pop-up. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable.</p> <p>The Affinity Groups pop-up (<code>AffinityGroupSearchPopup.pcf</code>) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the <code>affinitygroupadmin</code> permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab.</p> <p>See <i>Application Guide</i>.</p>
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</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. The drop-down list contains a configurable set of choices.</p> <p>For more information about segmentation, see the <i>Configuration Guide</i>.</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>
Currency	<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, <code>Account.PreferredCoverageCurrency</code>.</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, <code>Account.PreferredSettlementCurrency</code>. You can view and change the preferred settlement currency on the Account Summary screen.</p>

This screen includes both **Secondary Named Insured** and **Additional Named Insureds**. In general, you include one or the other but not both. You can remove the unused field from the user interface.

Dwelling screen for homeowners

The **Dwelling** screen separates dwelling information into two tabs (cards): **Details** and **Additional Interests and Insureds**. Most fields on the **Details** tab are required for rating. Fields are validated with a warning that allows you to move to other wizard steps. Only fields that are required before moving to another step are configured as required in the user interface and marked with an asterisk. Unpopulated fields that are required for rating produce errors at quote time. You must correct these errors before quoting can proceed. There can also be unpopulated fields required for binding or issuance which produce errors at these times.

Dwelling Details tab for homeowners

In homeowners, the **Details** tab collects information needed about the dwelling that is to be insured. The information does not vary much by policy type.

General

Specify general location information in the upper left section of the **Details** tab. The dwelling's location state drives coverage availability and other data rules.

Occupancy

In **Occupancy**, specify how the property is occupied. **Usage** and **Occupancy** are maintained in typelists.

Protection

These fields collect information related to protection of the dwelling.

Fire Protection Class code can be filled in by clicking **Autofill Class Code**. You can also manually enter or override **Fire Protection Class**.

Most of the fields are not marked as required in the user interface and so do not prevent moving to other wizard steps if left unanswered.

Hazards

In **Hazards**, specify hazards on the premises. For each hazard, select whether the type is **Property** or **Liability**. **Specific Hazard** and **Type** are maintained in typelists.

Animals

In **Animals**, specify whether there are animals or exotic pets on the premises. For each animal, select a type, breed, and bite history. Type and breed are maintained in typelists. The information does not vary by policy type.

Swimming pool

In **Swimming Pool**, specify information about the pool.

Additional Interests and Insureds tab for homeowners

In homeowners, the **Additional Interests and Insureds** tab allows you to add or remove additional interests and additional insureds. The **Interest Type** and **Type** fields are typelists. The typelists are **AdditionalInterestType.ttx** and **AdditionalInsuredType.ttx**, respectively.

Dwelling Construction screen for homeowners

The **Dwelling Construction** screen provides fields for specifying the required information about the dwelling construction, such as year built, type of construction, number of stories, and roof type. This information is segregated in a separate screen to enable easier integration with Insurance To Value (ITV) and workflow control systems. The fields are marked as not required in the user interface.

Coverages screen for homeowners

The **Coverages** screen displays fields for selecting coverages and specifying values and limits. It is divided into the following tabs (cards): **Coverages**, **Additional Coverages**, **Optional Coverages**, and **Conditions and Exclusions**. The **Coverages** tab is further divided into **Section I Coverages** and **Section II Coverages**.

The **Policy Type** and **Coverage Form** selected on the **Qualification** screen together with the state specified on the **Dwelling** screen determine the content of the **Coverages** screen.

Coverage categories, configured in Product Designer under **Policy Lines**→**Homeowners Line**→**Categories**, determine on which tab each coverage appears. “Homeowners coverage categories” on page 242 provides a description of each coverage category.

Coverages tab

The main **Coverages** tab displays core homeowners coverages, including those that are required for the selected policy type and state. Coverages include Section I and Section II coverages. Within these categories, the actual coverages depend on the selected policy type and state.

The implementation uses specialized coverage term modal input sets to implement special coverage term availability rules. You can examine these in Studio by opening `HOPMainCoveragesPanelSet.pcf`.

Additional Coverages tab

The **Additional Coverages** tab displays additional liability coverages and additional property coverages for all policy types. All of the coverages are required. Within each of these categories, the actual coverages depend on the selected policy type and state.

Optional Coverages tab

The **Optional Coverages** tab displays coverages in the Optional Coverages category whose existence is Electable in the product model. Coverages are grouped into **Policy Line Optional Coverages** and **Dwelling Optional Coverages**.

Conditions and Exclusions tab

The **Conditions and Exclusions** tab enables you to enter exclusions and conditions. In the base configuration of the product model, you can attach conditions and exclusions to the line or dwelling.

Modifiers screen for homeowners

The **Modifiers** screen enables you to indicate if a policyholder is eligible for a modifier (premium debit or credit). In the base configuration, you can specify modifiers on the policy or dwelling.

A modifier is a value used by the rating engine to adjust the policy premium or some portion of the premium.

Some modifiers capture information relevant to the pricing of a policy that is not necessarily tied to a specific coverable or coverage. However, under the **Dwelling** column, modifiers are specific to the dwelling. If you configure homeowners with multiple dwellings, you can capture modifiers for each dwelling.

Risk Analysis screen for homeowners

The **Risk Analysis** screen enables you to enter data that is used to evaluate the risk of the applicant and view risks that were identified during the submission process.

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.

- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.
- **Claims** – 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 784.
- **Prior Losses** – Displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

Policy Review screen for homeowners

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. For other policy transactions, this screen displays the differences between the policy versions. In a policy change, out-of-sequence conflicts appear on an **Out-of-Sequence** tab. 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.

The **Policy Review** screen displays coverages, exclusions, policy conditions, and modifiers. Where a value applies, the screen displays the value for each item. If PolicyCenter is configured as a multicurrency system, values are in the currency set on the coverable.

This screen includes conditions and exclusions selected under **Coverages**. Conditions and exclusions are typically documented by forms attached to the policy.

Quote screen for homeowners

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen. This screen 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

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 511.

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 in Rating Management” on page 542.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Policy Premium tab on Quote screen

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 411.

Disclosing base premium amounts is a requirement of homeowners insurance that is not applicable in most other lines of business. The base premium value is calculated on the base coverages for the policy, including the default limits and terms for those coverages. When non-default limits or terms are selected for base coverages, these adjustments to the base premium are listed separately.

Due to base premium disclosure requirements in homeowners policies, coverage term values listed in the **Base Premium** section of the quote are not necessarily the ones specified for the policy. Instead, the coverage term values are the default term values used in the base premium calculation.

Non-default terms and values, and their associated adjustments to premium are listed in the **Adjustments to Base** section. Additional coverages and their limits, terms, and associated premiums are listed in the **Other Premiums** section.

See also

- “Homeowners rating” on page 221

Cost Change Detail tab

In all job other than submission jobs, the **Quote** screen includes a **Cost Change Detail** tab. The information on this tab provides the offset and onset transactions associated with the job. The cost change subtotal appears at the bottom of the screen.

Forms screen for homeowners

The **Forms** screen displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 for a description of the PolicyCenter Forms Inference engine and how it works.

Payment screen for homeowners

The **Payment** screen displays payment information for the policy.

See also

- “Working with the Payment screen” on page 775

Field validation in homeowners

Most of the fields in the homeowners wizard are required for rating but relatively few are marked as required in the user interface. The user interface accommodates cases in which some of the required information is missing when you initially begin a policy submission job. In the base configuration, navigating away from a screen invokes field-level validation and warns you about fields that are required for rating. You can safely ignore these warnings and repeat the navigation to move to another screen. However, when you request a quote, field-level validation repeats to

ensure that all rating-required fields are populated. At this point, PolicyCenter displays errors for any missing required values. You must correct these errors before quoting can proceed.

Quote process for homeowners

The quote process for the homeowners line of business includes validation, rating, generating underwriting issues, and generating forms.

Quote-time validation for homeowners

As explained in “Field validation in homeowners” on page 228, the user interface does not require that you specify all rating-required fields before leaving the screen where they appear. Therefore, you need not have the answers for many of the questions required to obtain a quote during the initial data entry process. However, PolicyCenter performs validation on these rating-required fields and raises any errors prior to attempting to obtain a quote from the rating engine. PolicyCenter runs the same validations at quote time as it runs when exiting the **Dwelling** and **Dwelling Construction** screens. However, any issues raised at quote time result in errors instead of warnings. You must correct all errors must before rating can proceed.

Homeowners rating premium

In the base configuration of homeowners, Guidewire Rating Management implements very basic sample rating logic for only some of the defined coverages. This rating logic is implemented in the `gw.lob.hop.HOPRatingEngine` Gosu class. The implementation also includes rate books, rate tables, and rate routines. To implement rating fully, Guidewire expects insurers to use Guidewire Rating Management or integrate with an external rating engine.

The following coverages return sample premiums from the rating logic provided in the base configuration of Rating Management for homeowners:

- Coverages in the category “Section I Coverages”
- Coverages in the category “Section II Coverages”
- Homeowners Ordinance Or Law
- Scheduled Personal Property

Note: No other coverages return a premium.

The base configuration of Rating Management for homeowners includes rating for rate modifiers.

Homeowners configuration

In the base configuration, the homeowners line of business provides a starting point for your implementation. Review the content provided and determine the changes you need to make to suit your requirements. This topic describes how to configure the homeowners line of business provided in the base configuration.

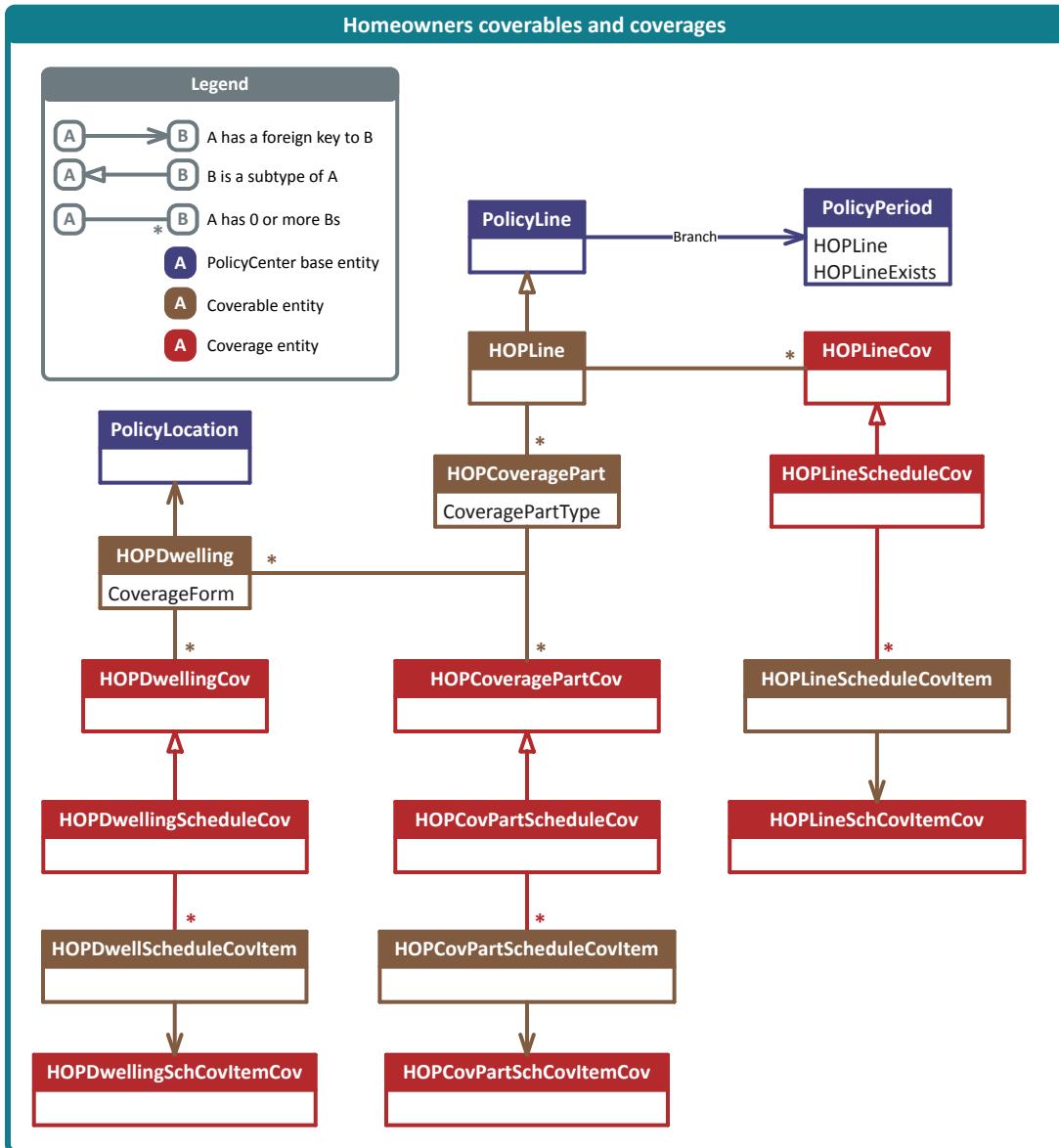
Homeowners data model

This topic describes the objects or entities associated with the homeowners 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.

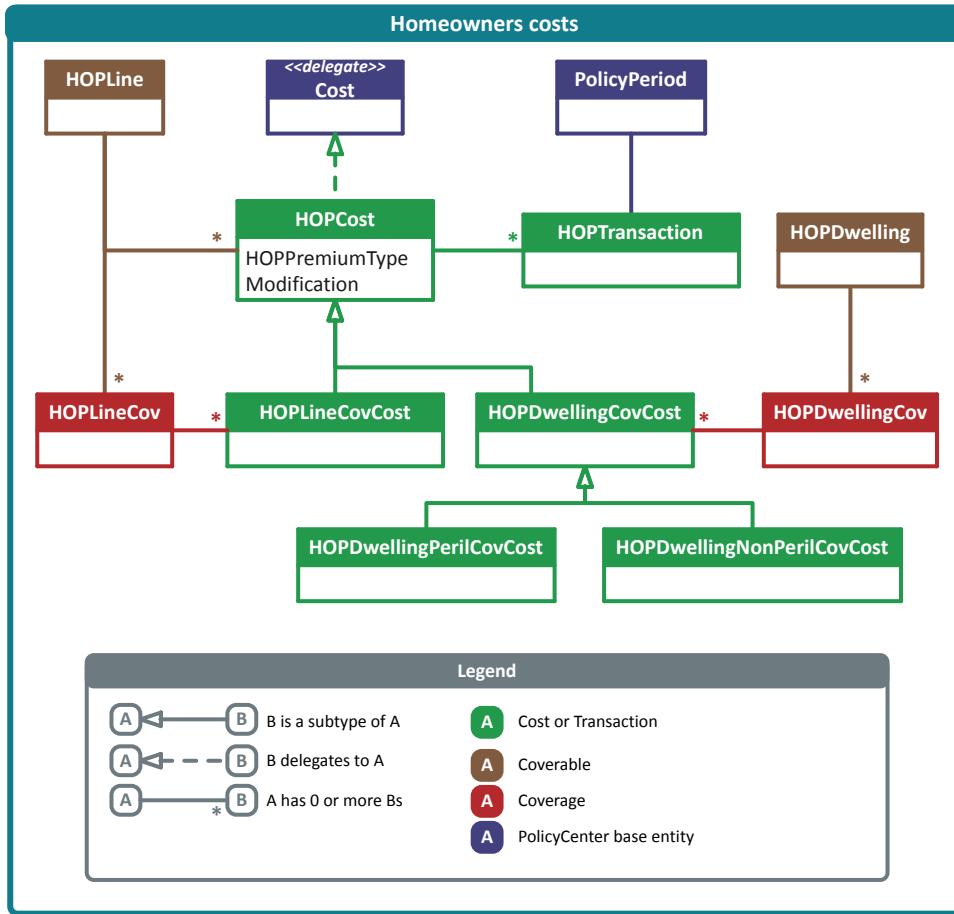
Homeowners coverables and coverages data model diagram

The following diagram shows entities related to coverables and coverages.

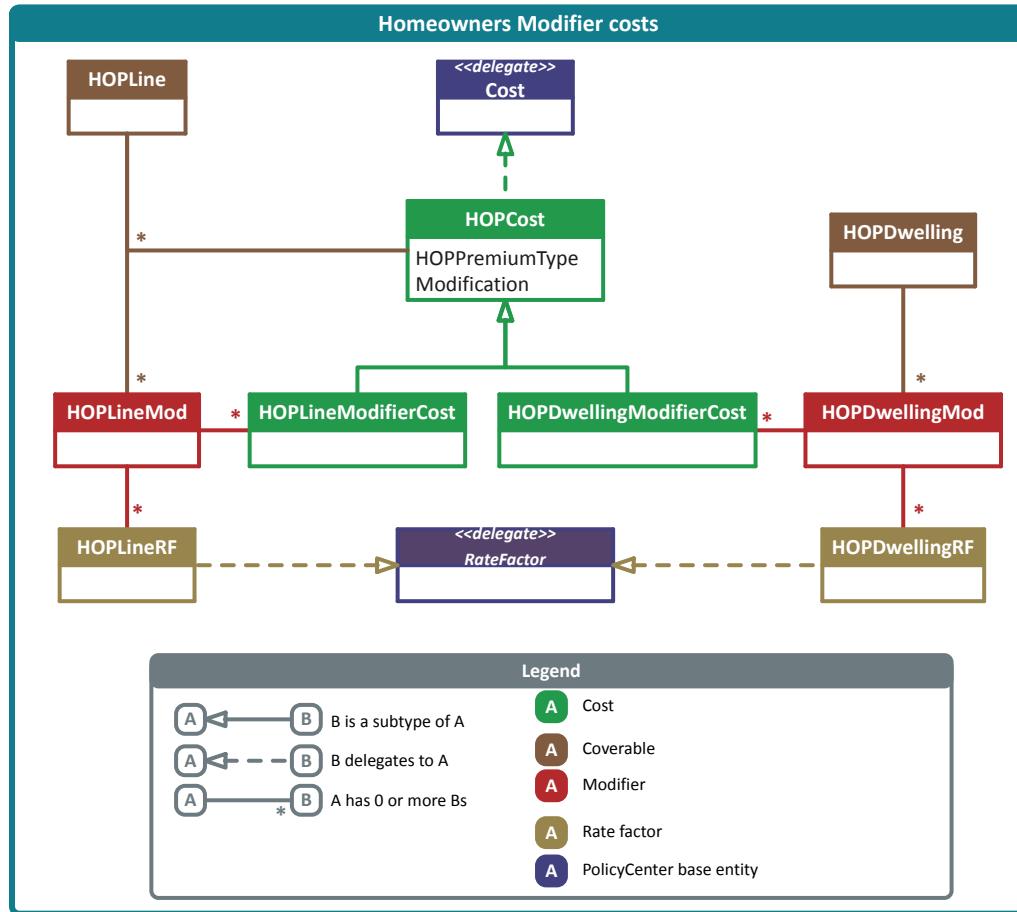


Homeowners cost data model diagram

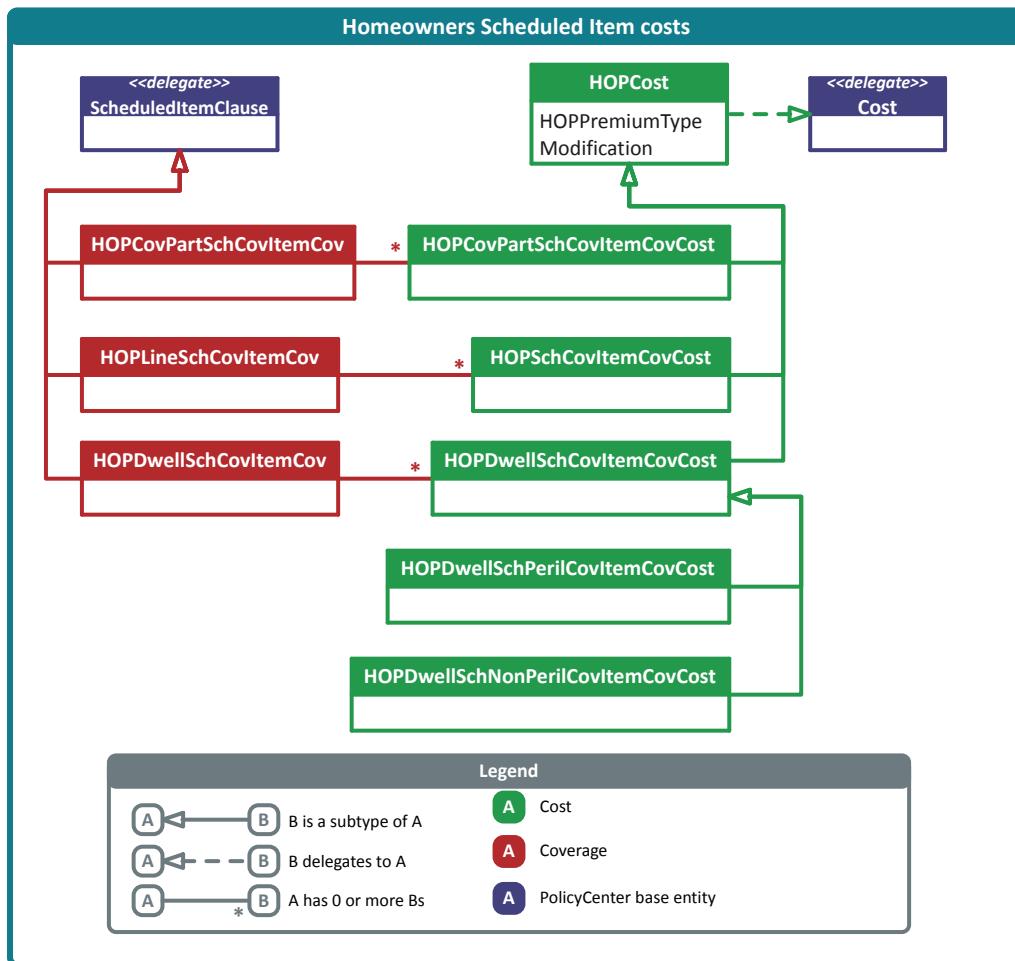
The following diagram shows entities related to costs, coverables, and coverages, and shows their relationships with other homeowners entities and with entities in base PolicyCenter.



The following diagram shows cost entities related to modifiers.

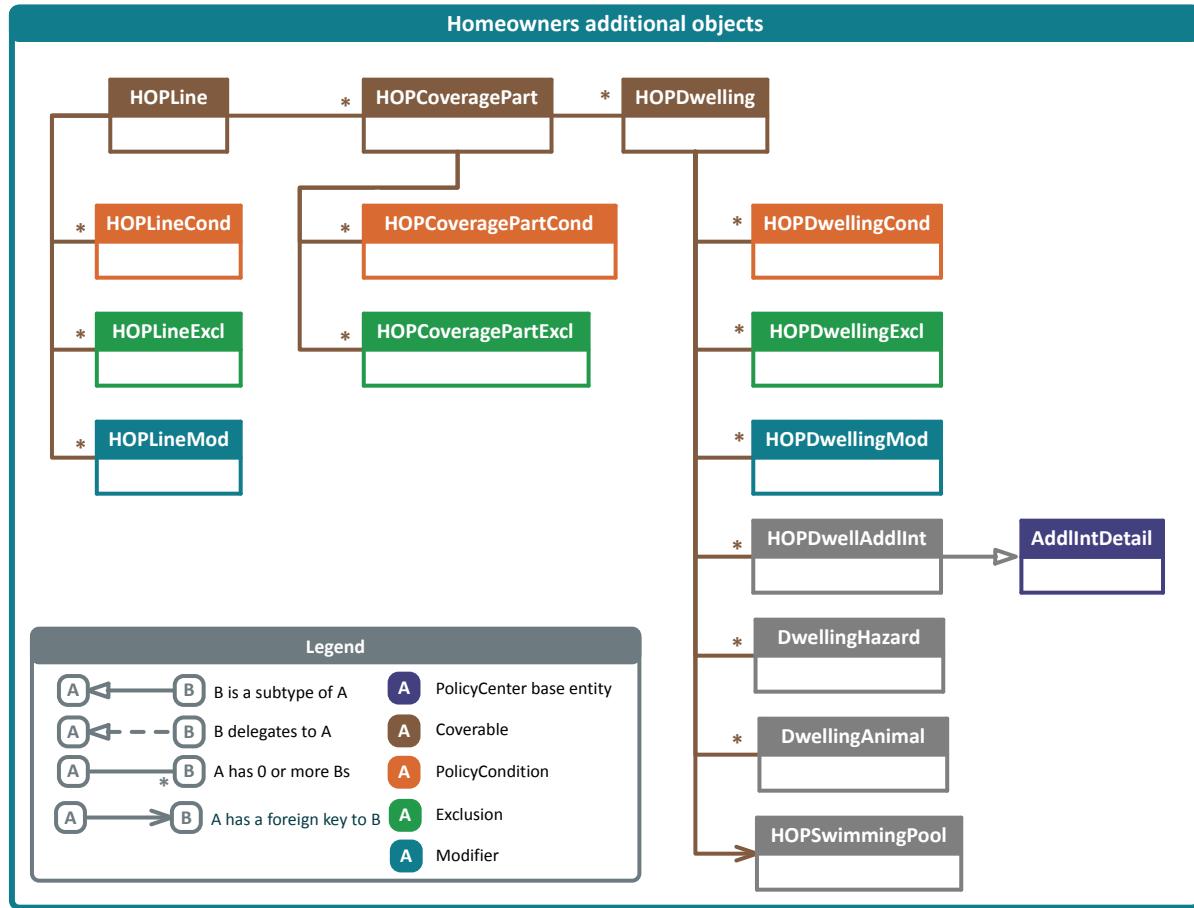


The following diagram shows cost entities related to scheduled items.

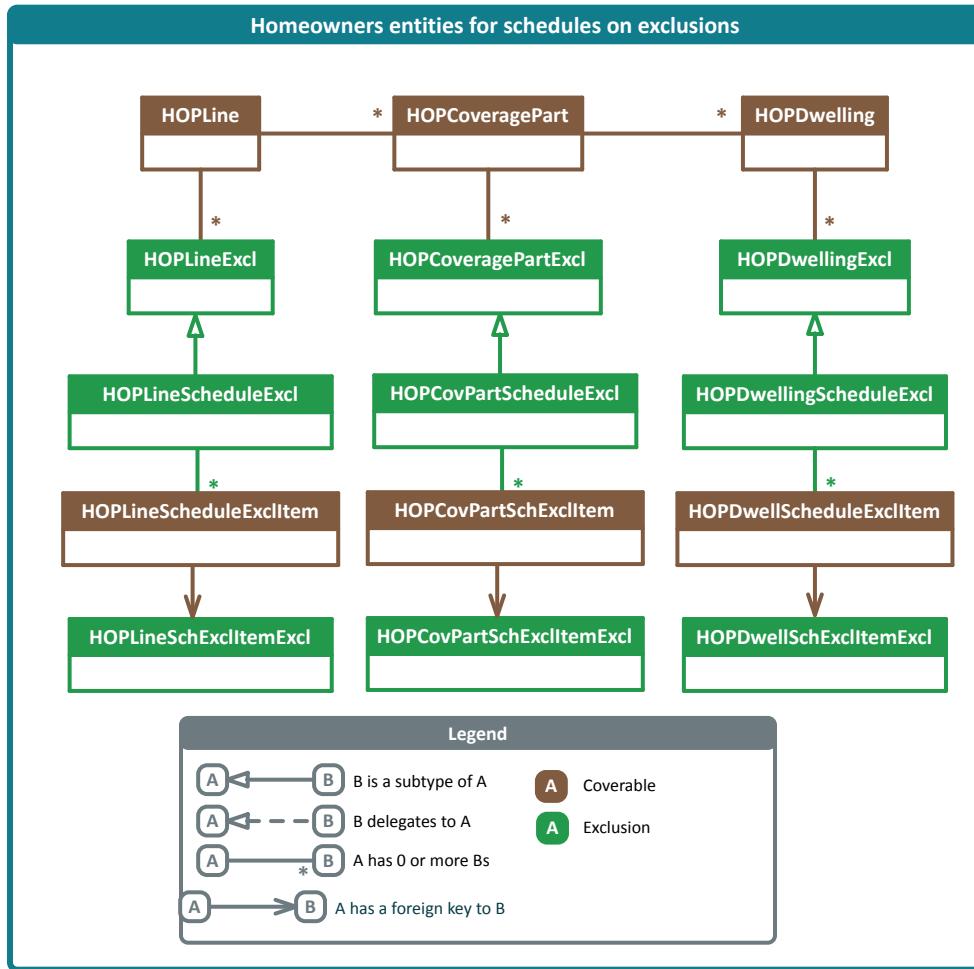


Additional homeowners coverables data model diagram

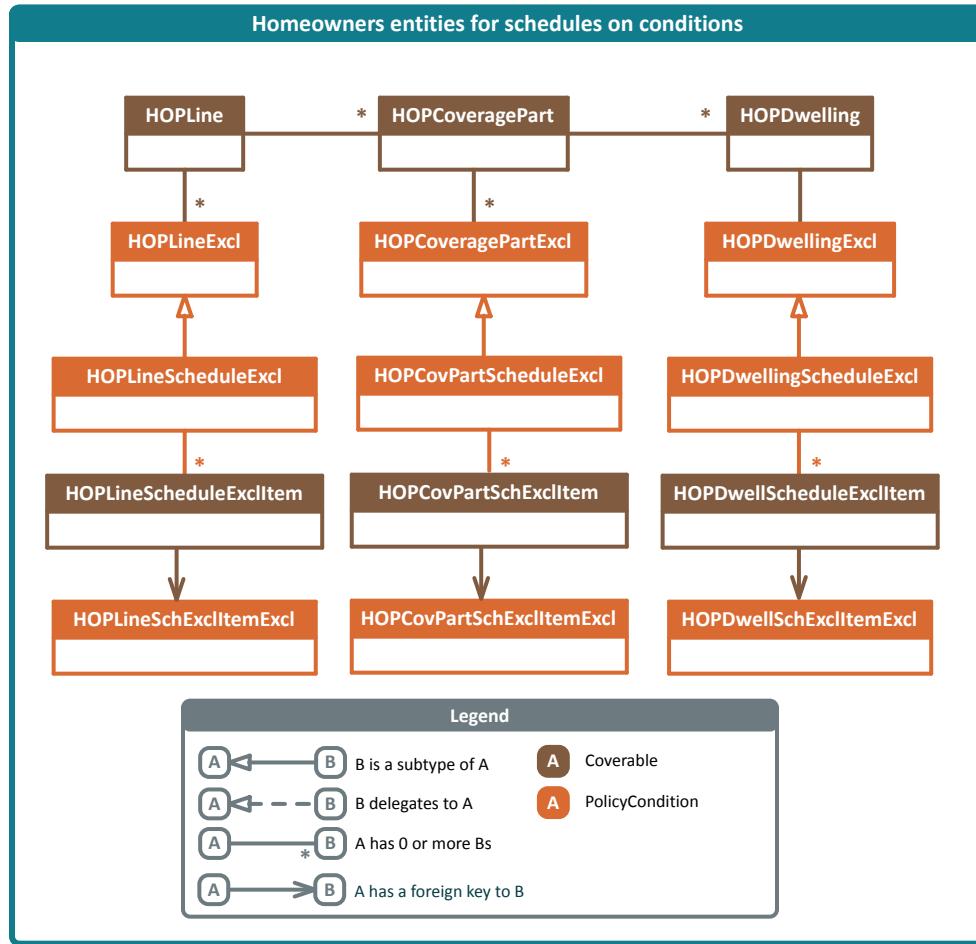
The following diagram shows additional entities related to coverables.



The following diagram shows entities related to schedules on exclusions.



The following diagram shows entities related to schedules on conditions.



Entities related to coverables in homeowners

The following entities are related to coverables in homeowners.

HOPLine

The HOPLine entity is a subtype of PolicyLine. The line delegates to Coverable and has an array to coverages, HOPLineCov entity instances. The line has arrays to conditions and exclusions, HOPLineCond and HOPLineExcl entity instances, respectively.

The line delegates to Modifiable and has an array of modifiers, HOPLineMod entity instances.

The line has arrays of PolicyAddlInsured and HOPCoveragePart entity instances.

HOPCoveragePart

In PolicyCenter, homeowners provides policy types for insuring dwellings, rentals, and condominiums. In the base implementation, the policy type is represented by the coverage part, HOPCoveragePart. In the user interface, the line has one and only one coverage part, therefore policy type and coverage part are somewhat interchangeable. For example, you have a dwelling and a rental property. You must get one policy to insure the dwelling and another policy to insure the rental property.

However, the data model supports an array of coverage parts on the line. Therefore, you can modify the implementation and the user interface to support multiple coverage parts. For example, one policy can include both the dwelling and the rental property.

Coverage parts can contain coverages, exclusions, conditions, modifiers, and coverable objects. However, clauses are often not specific to coverage parts or policy type. Clause availability may be determined by coverage part.

HOPDwelling

The HOPDwelling entity represents the dwelling covered by the homeowners policy. The dwelling references the location, `PolicyLocation` entity instance. The dwelling includes many details including protection information and construction details.

The `CoverageForm` property specifies the coverage form selection which can determine clause availability, such as coverage and coverage term availability. `CoverageForm` availability depends on the parent `HOPCoveragePart`. Coverage form selection can determine form inference.

HOPDwelling delegates to `Coverable` and holds an array of coverages (`HOPDwellingCov`) and an array of additional interests, `HOPDwellAddlInterest`.

HOPDwelling delegates to `Modifiable`, holding an array of modifiers. HOPDwelling also has arrays of conditions and exclusions.

Other entities

The coverage entities `HOPCoveragePartCov`, `HOPCovPartScheduleCov` and `HOPCovPartSchCovItemCov`, and the `HOPCovPartScheduleCovItem` coverable entity are provided for you to use and extend. In the base configuration, these entities are not associated with any coverage patterns.

Entities related to coverages in homeowners

The following entities are related to coverages in homeowners.

HOPLineCov

`HOPLineCov` is a coverage entity for homeowners coverages at the policy line level. `HOPLineCov` has an array of `Cost` (`HOPLineCovCost`) entities.

HOPLineScheduleCov

`HOPLineScheduleCov`, a subtype of `HOPLineCov`, defines homeowners line coverages that include schedules. This entity has an array of `ScheduledItem` (`HOPLineScheduleCovItem`) coverable entities. As a subtype of `HOPLineCov`, this entity can have an array or costs (`HOPLineCovCost` entity instances). `HOPLineSchCovItemCov` is the coverage entity associated with this coverable.

HOPLineScheduleCovItem

The `HOPLineScheduleCovItem` coverable entity defines a scheduled item of a homeowners line coverage. `HOPLineSchCovItemCov` is the coverage entity associated with this coverable.

This entity has an optional reference to a `PolicyLocation` entity. This reference can handle scheduled items that are not at the same location as the location of the dwelling.

HOPDwellingCov

`HOPDwellingCov` defines the types of coverage terms that can be assigned in the product model for homeowners dwelling-level coverages. `HOPDwellingCov` has an array of `Cost` (`HOPDwellingCovCost`) entities.

HOPDwellingScheduleCov

`HOPDwellingScheduleCov`, a subtype of `HOPDwellingCov`, defines dwelling coverages that include schedules. `HOPDwellingScheduleCov` has an array of `ScheduledItem` (`HOPDwellScheduleCovItem`) coverable entities.

HOPDwellScheduleCovItem

The `HOPDwellScheduleCovItem` coverable entity defines a scheduled item of a dwelling coverage. `HOPLineSchCovItemCov` is the coverage entity associated with this coverable.

This entity has an optional reference to a `PolicyLocation` entity. This reference can handle scheduled items that are not at the same location as the location of the dwelling.

Entities related to costs in homeowners

The following are entities related to costs in homeowners.

HOPCost

HOPCost is the super-type of all the cost entities in the homeowners line. HOPCost has an array of transaction (HOPTransaction) entities.

HOPCost rows include a type key for HOPPremiumType and Modification. These properties are distinctive for homeowners, particularly in the United States.

- **HOPPremiumType** – There are three premium types: Base Premium, Adjustment to Base Premium, and Other Premium. For United States markets, the base premium is calculated with the default coverage values for the given territory and jurisdiction. Then Adjustments to Base Premium are calculated for changes from the default, such as increased limits, changes in deductibles, among others. Other Premium is any other type of premium.
- **Modification** – It is a common practice in the United States to show the benefit of various policy discounts, such as multi-policy discounts or loyalty discounts. Therefore, the premium discount values (modifications) must be persisted in the cost rows. The undiscounted premium cost has a Modification set to Base, and the discounts are coded with Modification set to Modification.

HOPLineCovCost

HOPLineCovCost is a subtype of HOPCost for capturing line-level coverage costs.

Costs for scheduled items on the line can use HOPLineCovCost.

HOPDwellingCovCost

HOPDwellingCovCost is a subtype of HOPCost for capturing homeowners dwelling coverage costs. It is common that some rating for the dwelling is done by peril. To support this, HOPDwellingCovCost has subtypes for coverage costs that are rated by peril (HOPDwellingPerilCovCost) and non-peril rated items (HOPDwellingNonPerilCovCost).

Costs for scheduled items on the dwelling can use these cost subtypes.

Modifier costs

HOPLineModifierCost and HOPDwellingModifierCost are subtypes of HOPCost for capturing modifier costs.

Scheduled item costs

HOPCovPartSchCovItemCovCost, HOPLineSchCovItemCovCost, and HOPDwellSchCovItemCovCost are subtypes of HOPCost for capturing scheduled item costs.

HOPDwellSchPerilCovItemCovCost and HOPDwellSchNonPerilCovItemCovCost are subtypes of HOPDwellSchCovItemCovCost.

Homeowners typelists

The homeowners line uses the following typelists.

HOPCoverageForm typelist

The HOPCoverageForm typelist provides values such as H02 and H03 for **Coverage Form** in the **Qualification** screen.

You can add this typelist to the availability criteria for questions, coverages, coverage terms, and any other pieces of the product model that use availability. For example, conditions, exclusions, modifiers, question sets, and offerings, among others.

CoveragePartType typelist

The CoveragePartType.HOP typelist extension provides values for homeowners coverage parts.

HOPPremiumType typelist

For calculating the policy premium, this typelist defines different types of premium. In the base configuration, the types are:

- Base Premium
- Adjustment to Base Premium
- Other Premium

Modification typelist

For calculating the policy premium, this typelist indicates whether the cost is base (normal premium), or if it is a discount (modification) to the premium. The types are:

- Base
- Modification

Additional typelists for homeowners

Homeowners uses the following base typelists:

- | | | |
|--------------------------|------------------------|-----------------------|
| • AdditionalInterestType | • FoundationType | • HazardType |
| • AdditionalInsuredType | • FuelTankLocationType | • PlumbingType |
| • AnimalBreed | • FuelLineLocationType | • SmokeAlarms |
| • AnimalType | • GarageType | • SpecificHazard |
| • BurglerAlarmType | • HOPConstructionType | • SprinklerSystemType |
| • BreakerType | • HOPCoverageForm | • ResidenceType |
| • DwellingLocationType | • HOPPremiumType | • TermType |
| • DwellingOccupancyType | • HOPRoofType | • WindRating |
| • DwellingUsage | • HOPSwimmingPoolType | • WindType |
| • FireAlarmType | • HeatingType | • WiringType |

TermType is Annual or Other in the base configuration.

You can extend homeowners to access the WindRating and WindType typelists.

In the AdditionalInterestType typelist, typecodes specific to homeowners include a category with code starting in HOP.

In the AdditionalInsuredType typelist, typecode specific to homeowners include a category with code HOPLine.

Policy type and coverage form columns in homeowners

In homeowners, much of the content as well as the configuration of rules is closely tied to policy type. Some is also tied to coverage form. The policy type (dwelling, condominium, or rental) broadly defines pattern availability. The coverage form (HO2, HO3) further limits pattern availability.

In the homeowners line, pattern availability is based on *policy type*, whether the policy insures a dwelling, condominium, or rental. After selecting policy type, pattern availability is further narrowed by the selected *coverage form*. If the policy insures a dwelling, pattern availability is further narrowed down based on the level of insurance, specified as HO2, HO3, and HO5. For a rental policy, the only level is HO4.

Configuring the homeowners product model

You can configure the homeowners product model in Product Designer.

In the homeowners line, coverage categories drive much of the user interface, so give careful consideration to the category assigned to each coverage pattern. A coverage category controls how and where coverages appear in the user interface. You can use coverage category to define what appears in the primary workflow and what requires user actions to become visible or selectable.

The product model includes conditions, exclusion, and modifiers. Review these to ensure that they meet your requirements.

The product model includes scheduled coverages.

See also

- *Product Model Guide*

Homeowners coverage categories

In the homeowners user interface, coverages, conditions, and exclusions are grouped as Section I, Section II, Additional, Optional, among others. The groupings correspond to *coverage category*. In the product model, coverages, conditions, and exclusions have a **Category** field that specifies coverage category.

Section I and Section II coverages are groupings defined in the United States. Section I is for required property coverages and Section II is for liability coverages.

In the base configuration, homeowners provides a number of clause categories. Clause categories whose names start with **PC**, such as **PC Conditions**, appear only in PolicyCenter. The remaining clause categories appear in both PolicyCenter and Guidewire Digital Portals implementations.

Clause categories are used for grouping coverages, conditions and exclusions in the user interface. Change the coverage categories as needed to change how they are presented in the user interface.

For clause categories that only appear in PolicyCenter, the category code begins with the prefix **HOPPC**. The prefix enables filtering out these coverages from the Digital Portals user interface.

The coverage categories in homeowners are:

Category Code	Category Name	Description
Coverages tab		
HOPSectionICovCat	Section I Coverages	Coverage patterns based on industry standards for Section I base coverages. Coverages assigned to this coverage category appear in Section I Coverages .
HOPSectionIPCCovCat	Section I PC Coverages	Same as above.
HOPSectionIICovCat	Section II Coverages	Coverage patterns based on industry standards for Section II base coverages. Coverages assigned to this coverage category appear in Section II Coverages .
HOPSectionIIPCCovCat	Section II PC Coverages	Same as above.
Additional Coverages tab		
HOPAdditionalCovCat	Additional Coverages	Coverage patterns that are not base coverages in Section I or II but are typically required or suggested.
HOPPCAdditionalCovCat	PC Additional Coverages	Same as above.
Optional Coverages tab		
HOPOptionalCovCat	Optional Coverages	Coverage patterns that are not used for scheduled items and are defined as electable. The category includes coverages that have a schedule associated with them. Coverages assigned to this coverage category appear on the Optional Coverages tab. On the coverage pattern, the covered object, whether HOPLine or HOPDwelling , determines whether the coverage appears under Policy Line or Dwelling , respectively, on the Coverages screen. Additionally, the covered object type determines whether this coverage gets treated as coverage that has an associated schedule or not. For example, if HOPLine is the covered object, the covered object type is either HOPLineCov or HOPLineScheduleCov . If the covered object is HOPDwelling , then covered object type is either HOPDwellingCov or HOPDwellingScheduleCov .

Category Code	Category Name	Description
		The covered object can also be the HOPCoveragePart, but PolicyCenter does not display those coverages by default. To display these coverages, you can modify the PCF files.
HOPPCOptionalCovCat	PC Optional Coverages	Same as above.
HOPScheduledItemCovCat	Scheduled Item Coverages	<p>Coverage patterns that are used with scheduled items and are defined as electable.</p> <p>The covered object, whether HOPLineScheduleCovItem or HOPDwellScheduleCovItem, defined in the coverage pattern determines whether the coverage appears under Policy Line Optional Coverages or Dwelling Optional Coverages, respectively.</p> <p>Include all covered objects used for scheduled items (HOPLineScheduleItem, HOPDwellScheduleCovItem, and HOPCovPartScheduleCovItem) in this category.</p>
Conditions and Exclusions tab		
HOPConditions	Conditions	Coverage patterns defined for conditions. This coverage pattern category appears under Conditions . This category includes scheduled coverage patterns.
HOPPCConditions	PC Conditions	Same as above.
HOPExclusions	Exclusions	Coverage patterns defined for exclusions. This coverage pattern category appears under Exclusions . This category includes scheduled coverage patterns.
HOPPCExclusions	PC Exclusions	Same as above.

Use Product Designer to view details of coverage categories, coverages, limits, exclusions, and terms within the homeowners line.

Homeowners validation classes

In the base configuration, homeowners includes validation classes.

Homeowners includes the following validation classes:

- HOPLineValidation
- HOPDwellingValidation
- HOPCoveragePartValidation
- HOPCoveragesValidation

HOPLineValidation

The HOPLineValidation class performs the following validations:

- Only one coverage part per line
- Validates coverages
- Validates additional insureds

HOPDwellingValidation

The HOPDwellingValidation class performs the following validations:

- Dwelling location cannot be changed except on certain policy transactions

Because the base configuration supports only one dwelling per policy, the base state also cannot be changed on these policy transactions.

- A location protection class is required
- Validates additional interests
- Building upgrades must occur after the building date
- Garage not available except for dwelling
- Answers to qualification questions must be consistent with the characteristics of the related dwelling

[HOPCoveragePartValidation](#)

The `HOPCoveragePartValidation` class performs the following validations:

- Only one dwelling is permitted per coverage part

[HOPCovrancesValidation](#)

The `HOPCovrancesValidation` class performs the following validations:

- Various coverage checks. For example, building limit must not exceed replacement cost.
- Scheduled item limits must not exceed the personal property limit.

[See also](#)

- [Configuration Guide](#)

Configuring underwriting issues for homeowners

The base configuration provides examples of underwriting rules that create underwriting issues for homeowners. You can extend and modify the included rules to meet your requirements. Homeowners underwriting rule names begin with `HOP`.

Configuring forms for homeowners

The base configuration provides sample forms which demonstrate common patterns of inference for homeowners. Insurers are expected to configure their own forms for homeowners.

Homeowners integrations and plugins

Integrations typically performed when implementing PolicyCenter include rating, print/issuance, customer address book, claims, billing, and others. For homeowners, you may wish to integrate with functionality to obtain the following information:

- Protection class code
- Insurance to value (ITV), also known as Insured Value Report (IVR)

The workflow and design of the screens anticipates the need for such integrations.

Configuring homeowners rating

Homeowners includes two demonstration rating implementations:

- A Guidewire Rating Management implementation that provides a basic structure as well as examples for further development of rating. However, this is not a complete implementation of homeowners rating.
- A minimal Gosu-based implementation which is not suitable as a basis for further development. You can use this implementation for demonstration purposes. Prior to integrating with an external rating engine, this implementation enables you to bind and issue homeowners policies while working on other parts of PolicyCenter.

Guidewire Rating Management provides a rating system which includes a set of tools to manage and maintain rating in PolicyCenter.

In the base configuration, the code for the Gosu-based demonstration rating system is in the `gw.lob.hop.rating.HOPDemoRatingEngine` class.

In the Guidewire Rating Management implementation for homeowners, the rating implementation is in the `gw.lob.hop.rating.HOPRatingEngine` class. The implementation also includes rate books, rate tables, and rate routines.

See also

- “Homeowners rating” on page 221
- “Guidewire Rating Management for homeowners” on page 221
- *Integration Guide*

Protection class code for homeowners

On the **Dwelling** screen, you can enter a **Fire Protection Class** code. Click **Autofill Protection Class** to retrieve the protection class code from the protection class plugin (`IProtectionClassPlugin`) implementation. Protection class code is typically derived based on location attributes such as address, distance to fire station/hydrant, and others. The class code is often populated by calling an external service or database such as the ISO Location database. Therefore, setting the protection class code is implemented as a custom integration point.

In the base configuration, `MockProtectionClassPlugin.gs` implements the protection class plugin. This implementation is for demonstration and development purposes only.

Homeowners insurance to value

Many insurers integrate with service providers for insurance to value (ITV) calculations. Homeowners provides a separate **Dwelling Construction** screen to facilitate implementing the integration with a single screen.

You can customize input fields and data entry rules in the PCF to match the requirements of your service provider. You can add a button that manually triggers the service on demand. You also can add an automated interface call during the submission process, for example prior to quote or prior to bind. Upon return, set the value of the result fields to the values returned from the service. Values specify the limits. You can:

- Replace values.
- Compare values to drive underwriting rules and rating. This requires that you extend the data model and PCF files.

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 policy transactions, 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.

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, an insurer defines the coverage parts for which they will provide insurance. A typical insurer 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.

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
Date Quote Needed	PolicyCenter displays a validation message if this date is in the past.
Estimated Premium	An estimated amount that you can enter prior to quoting. For commercial lines of business in submission, renewal, or rewrite policy transactions. For renewals and rewrites, the value is populated from the value on the prior policy period. PolicyCenter updates the value with the Total Premium when the quote is released. It is also updated when the policy is bound or issued. The value also appears on the Policy Review screen.
Primary Named Insured	PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select: <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact 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. For Existing Contact , PolicyCenter lists account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured .
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. 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 This might be a business partner, for example.

Name of field	Description
Policy Details	Includes fields such as the Term Type , Effective Date , Expiration Date , and Written Date .
Affinity Group	The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search popup. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search popup. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable. The Affinity Groups popup (<code>AffinityGroupSearchPopup.pcf</code>) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the <code>affinitygroupadmin</code> permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab. <i>See Application Guide.</i>
Producer of Record	The Organization defaults to the producer that you selected on the New Submissions screen. You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change. Although you can change the values for Organization and Producer Code , PolicyCenter limits your choices by user permissions.
Producer of Service	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 . In a policy change, you can change the Producer Code .
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. For more information about segmentation, see the <i>Configuration Guide</i> .
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 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 view and change the preferred settlement currency on the Account Summary screen.

Coverage part selection screen for inland marine

On the **Coverage Part Selection** screen, you can select coverage parts for the policy. You must select at least one coverage part to create a valid policy. The base configuration has 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

On the **Buildings and Locations** screen, you can 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.
Coinsurance Pct.	Specify the coinsurance percentage.
Accts Receivable - 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	In this section, you can 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	In this section, you can 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 page of the policy line.

See also

- “Multicurrency features” on page 487

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 left 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.
Coinsurance	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	In this section, specify whether theft or vandalism is covered or excluded.
Unscheduled Equipment	In this section, 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	In this section, 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 base configuration, the questions are used to gather information but do not raise any underwriting issues. You can configure the questions to raise underwriting issues.

See also

- *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. The **Coinsurance** percentage applies to all signs.

Risk Analysis screen for inland marine

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.

- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.
- **Claims** – 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 784.
- **Prior Losses** – Displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

In the base configuration for inland marine, 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 411.

Policy Review screen for inland marine

This screen contains tabs for each coverage part added to the policy. Each tab contains a summary of the selected coverages.

For a submission policy transaction, this screen contains all the policy data in summary form. For other policy transactions, this screen displays the differences between the policy versions. In a policy change, out-of-sequence conflicts appear on an **Out-of-Sequence** tab. 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.

The **Policy Review** screen displays coverages, exclusions, policy conditions, and modifiers. Where a value applies, the screen displays the value for each item. If PolicyCenter is configured as a multicurrency system, values are 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 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

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 511.

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 in Rating Management” on page 542.

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 411.

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 displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 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.

See also

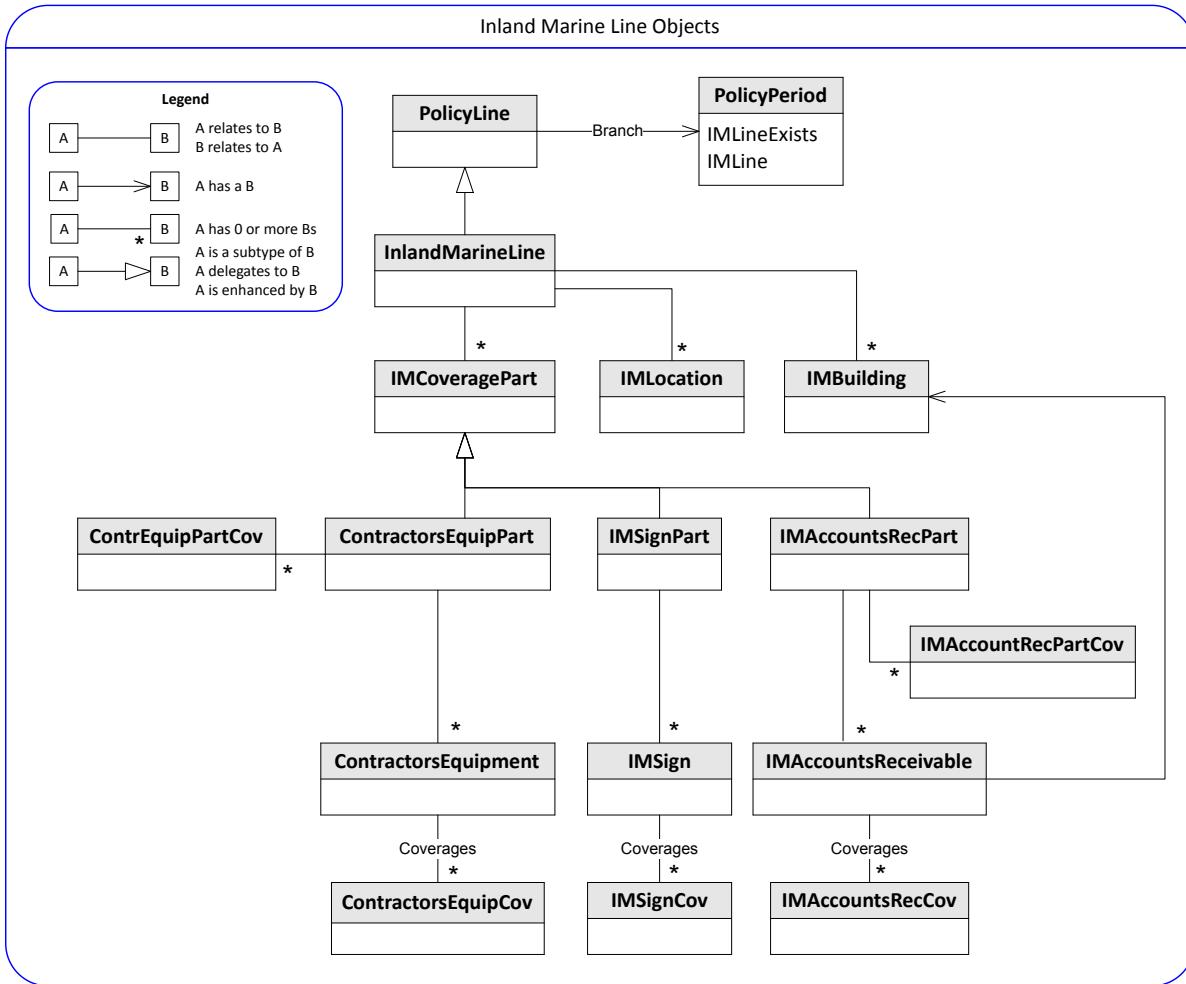
- “Working with the Payment screen” on page 775

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 300
- “Cost and transaction model for inland marine line” on page 403

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 Product Designer, the following items in the **Product Model** are related to inland marine:

- **Inland Marine** product
- **Inland Marine Line** policy line
- **IM Contractors Equipment Question** question set

See also

- *Product Model 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 policy transactions, 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.

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 insurer. 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 begins the initial capture of this information that is critical in obtaining a quote.

Motor vehicle records in personal auto

In personal auto, the 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.

An insurer 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

- *Integration Guide*

Motor vehicle record implementation overview

In personal auto, the Motor Vehicle Record (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 personal auto

In personal auto, 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 in personal auto” on page 268
- “Configuring personal auto motor vehicle records” on page 271

Working with personal auto

This topic contains step-by-step instructions for working with a personal auto policy in the user interface.

Copy coverages to other vehicles

About this task

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.

Procedure

- In a personal auto policy, navigate to the **PA Coverages** screen.
- Under the heading **Coverages applied per vehicle**, select **Copy Coverages**. This button is available if there are at least two vehicles on the policy.
- On the **Copy Coverages** screen, select a vehicle from the **Copy From Vehicle** drop-down list.
- 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 271

Personal auto screens

The personal auto policy transaction wizard steps capture information 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**)

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 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 account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured.</p>
Secondary Named Insured	<p>Use to create the secondary named insured. The drop-down menu enables you to select:</p> <ul style="list-style-type: none"> • New Person • From Address Book • Existing Contact <p>Secondary named insured might be a spouse or a child, for example.</p>
Affinity Group	<p>The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search popup. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search popup. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable.</p> <p>The Affinity Groups popup (AffinityGroupSearchPopup.pcf) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the affinitygroupadmin permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab.</p> <p>See <i>Application Guide</i>.</p>
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p>
Producer of Service	<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> <p>In a policy change, you can change the Producer Code.</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. The drop-down list contains a configurable set of choices.</p> <p>For more information about segmentation, see the <i>Configuration Guide</i>.</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>
Coverage	<p>The preferred or default currency for coverages on the policy. The 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>

Name of field	Description
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 view and change the preferred settlement currency on the Account Summary screen.</p>

Drivers screen for personal auto

On the **Drivers** screen, you can define the drivers of vehicles that this policy covers. Initially there are no drivers on the policy. 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 personal auto **Vehicles** screen, you create the vehicles that this policy insures. 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 36 for additional information.
Garaged At	The default in 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 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 page of the policy line.

See also

- “Multicurrency features” on page 487

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 insurers regarding auto policies.

The **PA Coverages** 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 check box 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 always appear. PolicyCenter displays these coverages initially, but they are not selected. Towing and Labor is an example of this kind of electable coverage.

Coverages applied per vehicle section

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 click **Copy Coverages** to copy coverages from one vehicle to others. For step-by-step instructions, see “Working with personal auto” on page 259.

Additional Coverages tab

On the **Additional Coverages** tab, you can choose other coverages that are less commonly selected, such as electronic equipment.

Risk Analysis screen for personal auto

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.
- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.

- **Claims** – 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 784.
- **Prior Losses** – Displays information about prior losses incurred by the insured. You can manually enter or attach information about prior losses.

Motor Vehicle Records tab

Risk Analysis in personal auto includes the **Motor Vehicle Records** tab. For each driver on the policy, this tab displays:

- **MVR Status**
- Summary data such as **Accidents**, **Violations**, and **Points**
- **Do Not Order MVR** status

The **Motor Vehicle Record** tab displays the MVR reports after completing the **All ordered MVRs received** activity.

Policy Review screen for personal auto

For a submission policy transaction, this screen contains all the policy data in summary form. For other policy transactions, this screen displays the differences between the policy versions. In a policy change, out-of-sequence conflicts appear on an **Out-of-Sequence** tab. 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.

The **Policy Review** screen displays coverages, exclusions, policy conditions, and modifiers. Where a value applies, the screen displays the value for each item. If PolicyCenter is configured as a multicurrency system, values are 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 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

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 511.

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 in Rating Management” on page 542.

When multicurrency display is enabled, the **Quote** screen displays all amounts in the settlement currency.

Forms screen for personal auto

The **Forms** screen displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 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.

See also

- “Working with the Payment screen” on page 775

Personal auto object model

This section describes the objects or entities associated with the personal auto line of business.

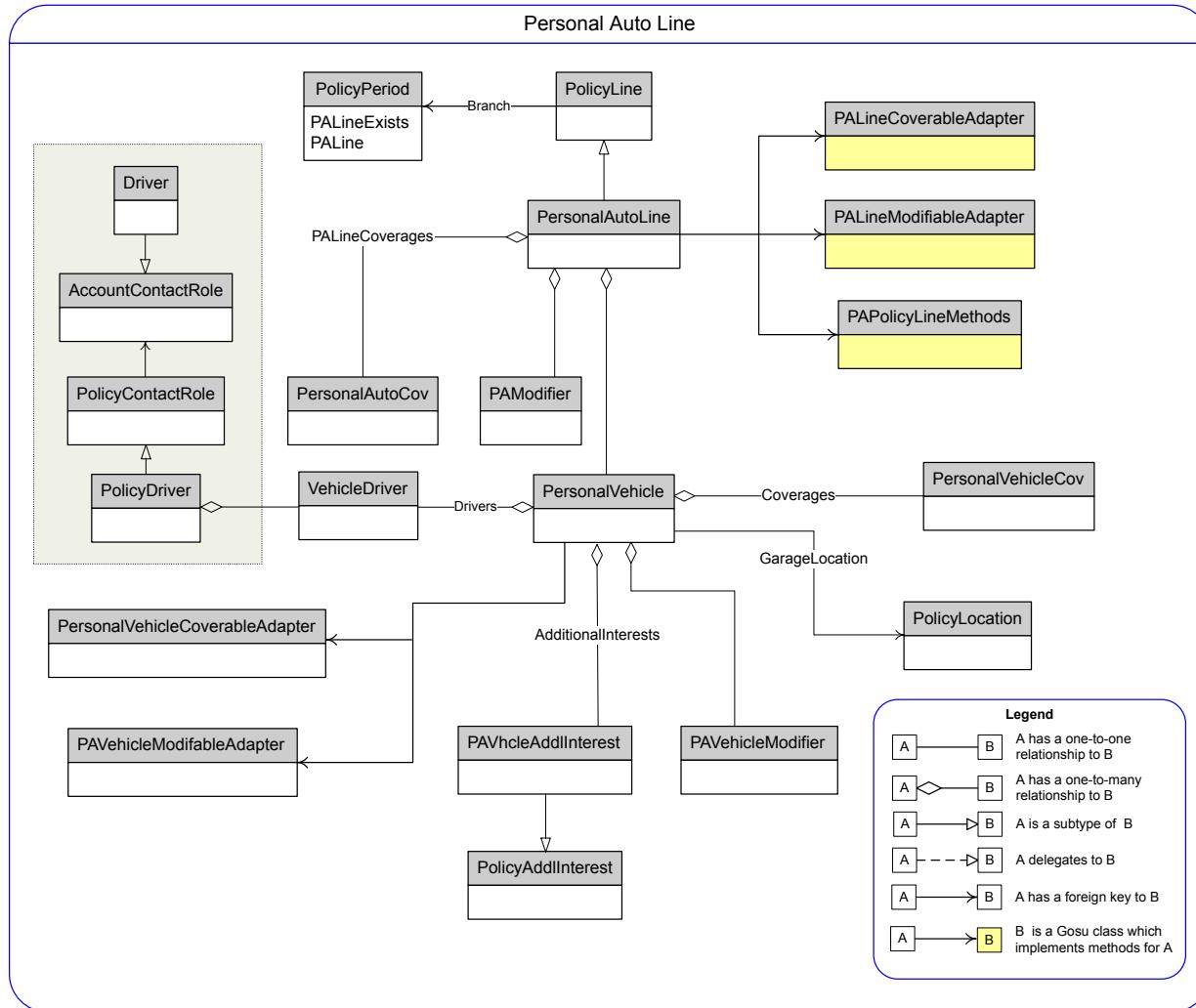
PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.

See also

- “Core entities associated with policies” on page 300
- “Cost and transaction model for personal auto line” on page 404

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 in personal auto

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 “Coverages, exclusions, conditions, and coverables overview” on page 447.

VehicleDriver entity in personal auto

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 in personal auto

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 in personal auto

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 in personal auto

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 in personal auto

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.

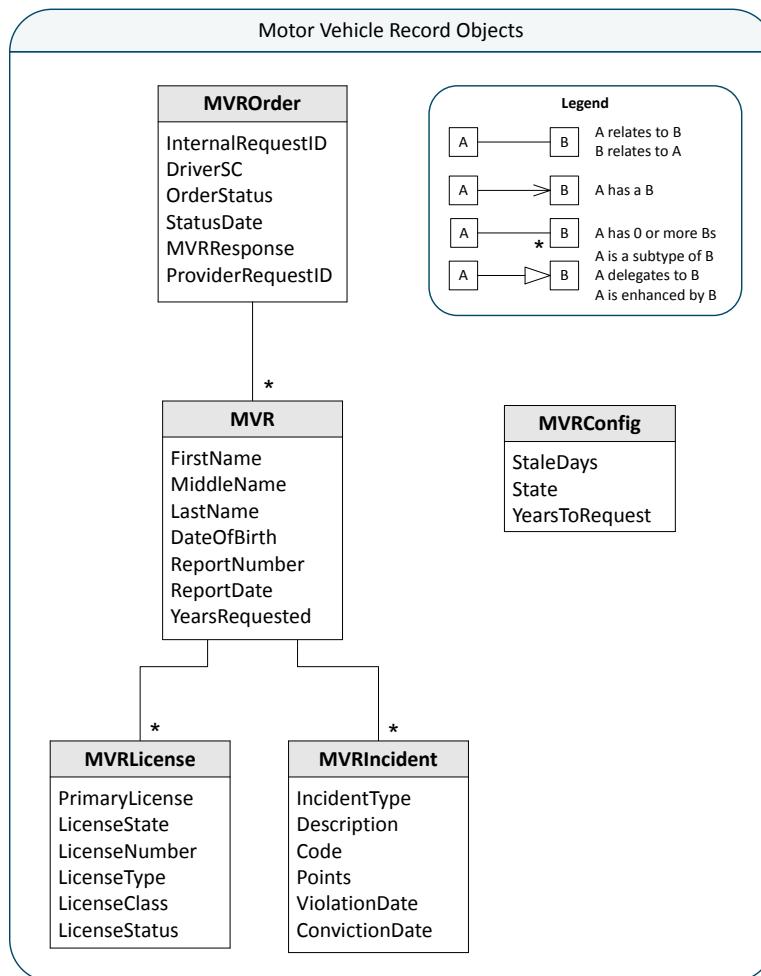
To learn more about how costs work in the personal auto line, see “Quoting and rating” on page 393.

Motor vehicle record object model in personal auto

In personal auto, 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:

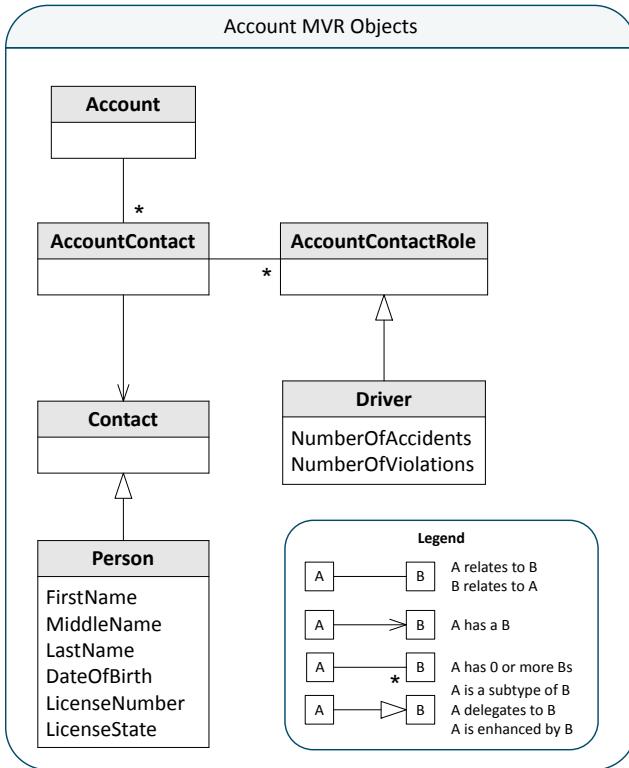
- **LicenseNumber**
- **LicenseState**
- **FirstName**
- **LastName**
- **MiddleName**
- **DateOfBirth**

The `gw.plugin.motorvehiclerecord.MVRSearchCriteria` class defines that fields that must match.

Account motor vehicle record object model in personal auto

In personal auto, Motor Vehicle Record (MVR) objects 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.

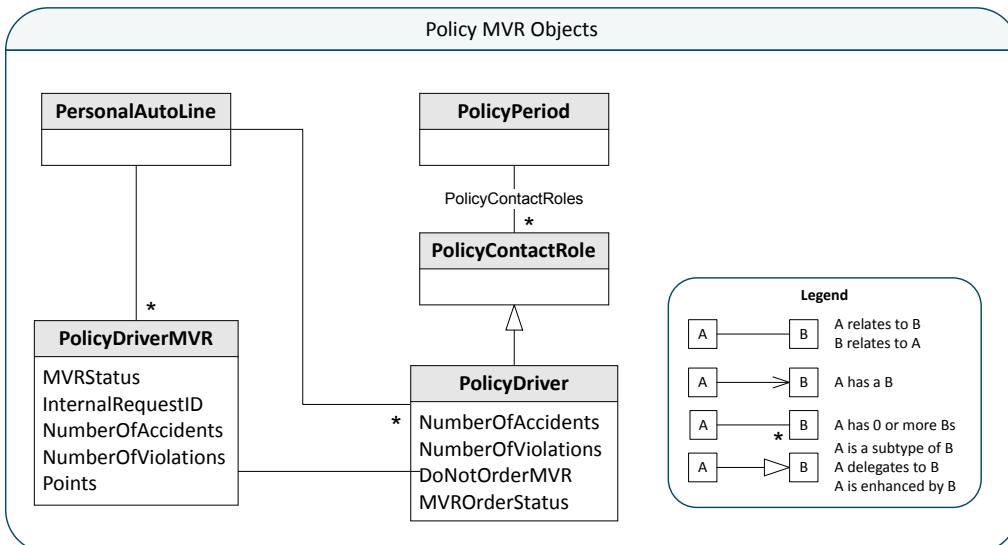
MVR data at the account level contains the latest number of accidents and violations entered on the **Drivers→Driver Details→Roles** tab in a policy transaction. The number of accidents and violations is stored on the **Driver** entity. These values are used on the next new term for that contact.



Policy motor vehicle record object model in personal auto

When an MVR is ordered in personal auto, 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 `copyCovernages` 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 “Motor vehicle record object model in personal auto” on page 268.

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.

In Product Designer, you can use the `motor_vehicle_record_configs.xml` system table to configure the criteria that PolicyCenter uses to search for motor vehicle records. This system table has the following columns:

Column	Description
<code>Jurisdiction</code>	The jurisdiction.
<code>UWCompanyCode</code>	The NAIC code (NAICCode) for the underwriting company. Specify this code on the <code>UWCompany</code> system table (<code>underwriting_companies.xml</code>) in Product Designer.
<code>YearsToRequest</code>	Number of years to search backwards for a MVR. If 7, search backward 7 years from the current date.
<code>StaleDays</code>	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 `Jurisdiction` and `UWCompanyCode` of the policy to find values for `YearsToRequest` and `StaleDays`. PolicyCenter uses the first match found in the `motor_vehicle_record_configs.xml` system table in the following order:

1. Both `Jurisdiction` and `UWCompanyCode` match
2. `Jurisdiction` matches and `UWCompanyCode` is not specified
3. `UWCompanyCode` matches and `Jurisdiction` value is not specified
4. `Jurisdiction` and `UWCompanyCode` are not specified

Note: If there are multiple rows with the same values for `Jurisdiction` 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

- 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 in Product Designer.

The checking set, `MVR`, 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 issues” on page 671
- *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 policy transactions, 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.

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.

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 139.

Policy term in workers' compensation

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 278.

Jurisdictions in workers' compensation

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 in workers’ compensation” on page 280.

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` system table in Product Designer. 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.

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 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 435.

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.

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 in workers' compensation" on page 277.

See also

- *Product Model Guide*

Split rating periods in workers' compensation

In workers' compensation, you can use split rating periods to create separate rating periods for dates selected by the user. You can create rating periods for each jurisdiction around one or more split dates.

Split rating periods are used to provide rating accuracy or to capture important information. In both cases, premiums are calculated separately for each period.

- Rating accuracy – Workers' compensation is a basis-rated line of business. If rating changes in the middle of a term, the basis is split to insure the accuracy of the rating. The basis amounts must be reported separately for each period. For example, this can occur when an experience modifier is revised.
- Capture information – Splits are used to determine the total premium before and after a major event of the insured. For example, an employer wants to split the premiums on the date of a bankruptcy or on the effective date that the company is sold a subsidiary. Even if not legally required, these splits may be important to the insurer or to the employer for tracking purposes.

Split for experience modifier corrections

When Unit Statistical Reports are corrected by the state rating authority, such as NCCI, for prior policy terms the correction may trigger a recalculation of the employer's experience modifier. You can use **Late Modifier** for this type of split. If the recalculation determines that the value of the experience modifier is lower, from 1.03 to .99 for example, the correction is reported to the current insurer as effective on the policy effective date. In this case, splitting the rating period is not required. If however the recalculation determines that the value of the experience modifier is higher, from .99 to 1.03 for example, the new corrected experience modifier is effective on the recalculation date, not the policy effective date. Since rating the policy requires including two different modifications, a split in basis is required on the effective date of the newer modifier.

Split for mandatory rate changes

Sometimes a jurisdiction issues a new set of rates amending the rates to be used within in-force policies. Since these new rates are effective for in-force periods, a rating period split is required to calculate the premium. You can use **Forced Rerating** for this type of split. A jurisdiction may issue these sudden and mandatory changes for various reasons, such as new legislation which significantly impacts the incurred amounts for claims.

Split for other rating reasons

You can use split rating periods for other types of midterm changes that affect basis-rated premium. For example, if the insurer allows midterm changes in employer liability limits, you can use a split rating period to apply the correct increase limits modifier to each period. You can create a reason code for this.

Split for informational reasons

In PolicyCenter, you can split rating periods for informational reasons. You can create a reason code for this.

Split for anniversary rating date

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.

Note: Split for anniversary rating date was eliminated by the National Council on Compensation Insurance (NCCI) in 2017.

The anniversary rating date is set on the jurisdiction. When an anniversary rating date designates that the policy must be split, PolicyCenter splits the rating period that contains that date 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 Rating Period** in Product Designer.

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 workers’ compensation” on page 283.

Inclusions and exclusions in workers' compensation

Workers' compensation in 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 in workers' compensation

Workers' compensation in 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 insurer always keeps.
- Loss Conversion Factor – A factor which is applied to losses when calculating dividends.

Specialty operations in workers' compensation

Workers' compensation in 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 insurer to prevent the insurer 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 insurer 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

The **Qualification** screen contains a set of questions to pre-qualify the applicant.

The questions reflect risks that the insurer 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 base 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 issues” on page 671
- *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
Date Quote Needed	PolicyCenter displays a validation message if this date is in the past.
Estimated Premium	An estimated amount that you can enter prior to quoting. For commercial lines of business in submission, renewal, or rewrite policy transactions. For renewals and rewrites, the value is populated from the value on the prior policy period. PolicyCenter updates the value with the Total Premium when the quote is released. It is also updated when the policy is bound or issued. The value also appears on the Policy Review screen.
Primary Named Insur- ed	PolicyCenter defaults the Primary Named Insured to the account holder. The Change To menu enables you to select: <ul style="list-style-type: none"> • New Company • New Person • From Address Book • Existing Contact 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. For Existing Contact , PolicyCenter lists account contacts that are account holder or named insured on the account. The list does not include contacts who are already the primary or additional named insureds on this policy. The listed contacts have an AccountContactRole of AccountHolder or NamedInsured .
Business and Operations	Designate if the policy is assigned risk. For Assigned Risk the default value is No . Assigned Risk may be set to Yes if an insurer directly writes or services this market segment and the submission qualifies as an assigned risk. If an insurer does not service or write directly in the non-standard market, change the configuration so that this element is hidden.

Field name	Description
	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.
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. 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 <p>This might be a business partner, for example.</p>
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 Policy Terms on the Workers' Compensation product in Product Designer.</p>
Affinity Group	<p>The Affinity Group section has a Name field with a search icon. The search icon displays the Affinity Group Search popup. You can type an affinity group name directly into the Name field or search for applicable affinity groups using the search popup. When you leave the Policy Info screen, validation ensures that the specified affinity group is acceptable.</p> <p>The Affinity Groups popup (AffinityGroupSearchPopup.pcf) immediately displays the set of acceptable affinity groups when it appears. Acceptable affinity groups are those whose definition does not preclude them from applying to the current policy. This screen enables you to search for an affinity group by name or type. In addition, if you have the affinitygroupadmin permission, you can click the name of an affinity group represented as a link in the search results. Clicking this link jumps directly to the Affinity Group editing screen in the Administration tab.</p> <p>See <i>Application Guide</i>.</p>
Producer of Record	<p>The Organization defaults to the producer that you selected on the New Submissions screen.</p> <p>You can change the Producer of Record in a rewrite or renewal. You cannot change the Producer of Record in a policy change.</p> <p>Although you can change the values for Organization and Producer Code, PolicyCenter limits your choices by user permissions.</p>
Producer of Service	<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> <p>In a policy change, you can change the Producer Code.</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. The drop-down list contains a configurable set of choices.</p> <p>For more information about segmentation, see the <i>Configuration Guide</i>.</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 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

Field name	Description
<ul style="list-style-type: none"> • 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 view and change the preferred settlement currency on the Account Summary screen.</p>	

Locations screen for workers' compensation

On 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.

Proof of coverage filings

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)	Standard industry classification. 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.

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 in workers' compensation

The **State Coverages** tab displays the jurisdictions covered on the policy. Clicking the jurisdiction in the list view opens the associated **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

In PolicyCenter, you can 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 January 1 of the current year. You set the **Anniversary Date** to July 1 of the previous year. PolicyCenter creates two rating periods: one from January 1 to July 1 of the current year and another from July 1 of the current year to January 1 of the following year.

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 base 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 the *Product Model Guide*.

Workers' Comp 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 the *Product Model Guide*. You can view the ID formats in the `official_id_validation_info.xml` system table in Product Designer.

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 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 "Jurisdictions in workers' compensation" on page 274. These options are configured in the <code>SpecialCov</code> 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.

Field name	Description
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 .
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.

Class Code Search screen

In workers' compensation, 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.

Policy Coverages and Exclusions tab in workers' compensation

In the **Policy Coverages and Exclusions** tab, you can select policy level coverages. Required coverages for the jurisdiction cannot be deselected.

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.
Statutory Workers' Comp	Indicates workers' compensation coverage as required by the covered jurisdictions. No additional coverage terms apply to this coverage.
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.

Field name	Description
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 multi-part limits are: <ul style="list-style-type: none"> • Per accident • Disease per employee • Disease per policy
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 monopolistic states • Listed states only • None
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`.

In Product Designer, you can configure policy coverages in the **Coverages** page in the **Workers' Comp Line** policy 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 the *Product Model Guide*.

Supplemental screen for workers' compensation

The **Supplemental** 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 `WCSupplemental` question set configured in Product Designer.

WC Options screen for workers' compensation

While workers' compensation has relatively few coverages, it has many contractual provisions with extended information needs. In the **WC Options** screen, you can select the options required for a given policy.

You can view this screen in Studio by navigating to `CoverageOptionsScreen.pcf`.

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 in workers' compensation” on page 280.

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 an insurer 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 insurer 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

On the **Risk Analysis** screen, you can enter data that is used to evaluate the risk of the applicant. You can also view risks that were identified during the policy transaction.

PolicyCenter passes this information to the rating engine. This screen has the following tabs to enter risk data:

- **UW Referral Reasons** – Appears when viewing a policy. It does not appear in a policy transaction 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 “Add underwriting referral reasons” on page 674.

- **UW Issues** – Displays underwriting issues that were raised during the job. For more information on this tab, see “Working with underwriting issues” on page 671.
- **Contingencies** – Displays contingencies associated with the policy or policy transaction. You can also add contingencies. Use *contingencies* to manage additional work on the policy and to take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For more information, see “Contingencies” on page 349.
- **Prior Policies** – Displays information about prior policies usually with another insurer. You can add information about prior policies.
- **Claims** – 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 784.
- **Prior Losses** – 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

For a submission policy transaction, this screen contains all the policy data in summary form. For other policy transactions, this screen displays the differences between the policy versions. In a policy change, out-of-sequence conflicts appear on an **Out-of-Sequence** tab. 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.

The **Policy Review** screen displays coverages, exclusions, policy conditions, and modifiers. Where a value applies, the screen displays the value for each item. If PolicyCenter is configured as a multicurrency system, values are in the currency set on the coverable.

Quote screen for workers’ compensation

Selecting the **Quote** button generates a quote and displays it on the read-only **Quote** screen. This screen 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

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 511.

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 in Rating Management” on page 542.

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 411.

Forms screen for workers' compensation

In the PolicyCenter base configuration, the workers' compensation line of business uses the base state for inferring forms.

The **Forms** screen displays the forms that have been inferred by forms inference.

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 in PolicyCenter.

In the base application, PolicyCenter identifies the forms to add:

- When quoting the policy
- When binding the policy

See also

- “Policy forms” on page 435
- “Policy form pattern administration” on page 745 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.

See also

- “Working with the Payment screen” on page 775

Audits

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.

See also

- *Integration Guide*

Workers' compensation object models

This section describes the objects or entities associated with the workers' compensation line of business.

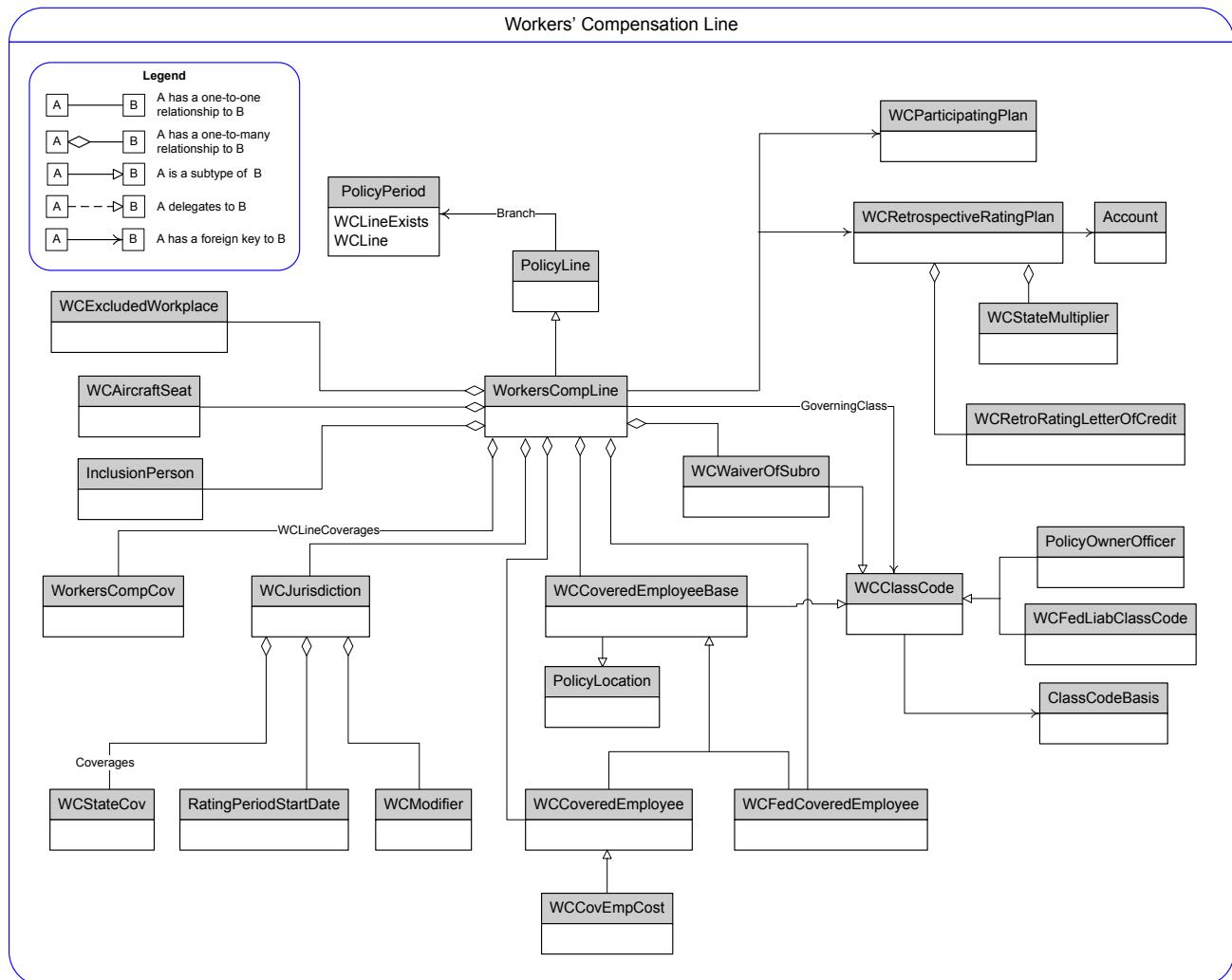
PolicyCenter provides a data model with entities or objects tailored to a particular line of business rather than a generic data model for all lines of business. Because PolicyCenter tailors entities to the line of business, they are easy to understand and work with. The coverage data objects for a line exist in their own database table, so you can make a change to one line of business without affecting the other lines.

See also

- “Working with policies” on page 298
- “Cost and transaction model for workers' compensation line” on page 406

Workers' compensation object model overview

The following diagrams show the main objects, or entities, of the workers' compensation business line and relationships between these entities.



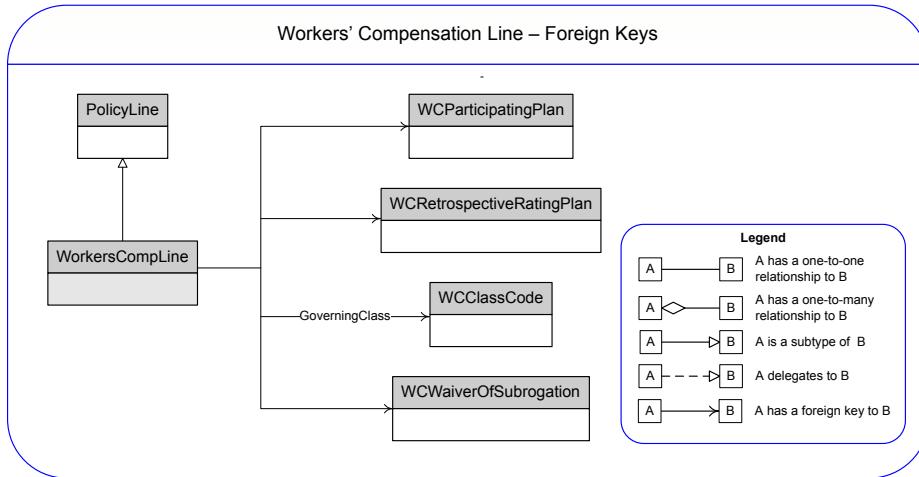
One-to-many entity relationships on WorkersCompLine

The the WorkersComplLine entity has a one-to-many relationship to:

- WCACoveredEmployeeBase
 - WCACoveredEmployee
 - WCFedCoveredEmployee
 - WCWaiverOfSubrogation
 - WCJurisdiction
 - WCExcludedWorkplace
 - WCAircraftSeat
 - InclusionPerson

Foreign keys on WorkersCompLine

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*.

Coverage entities in workers' compensation

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.

Modifier entity in workers' compensation

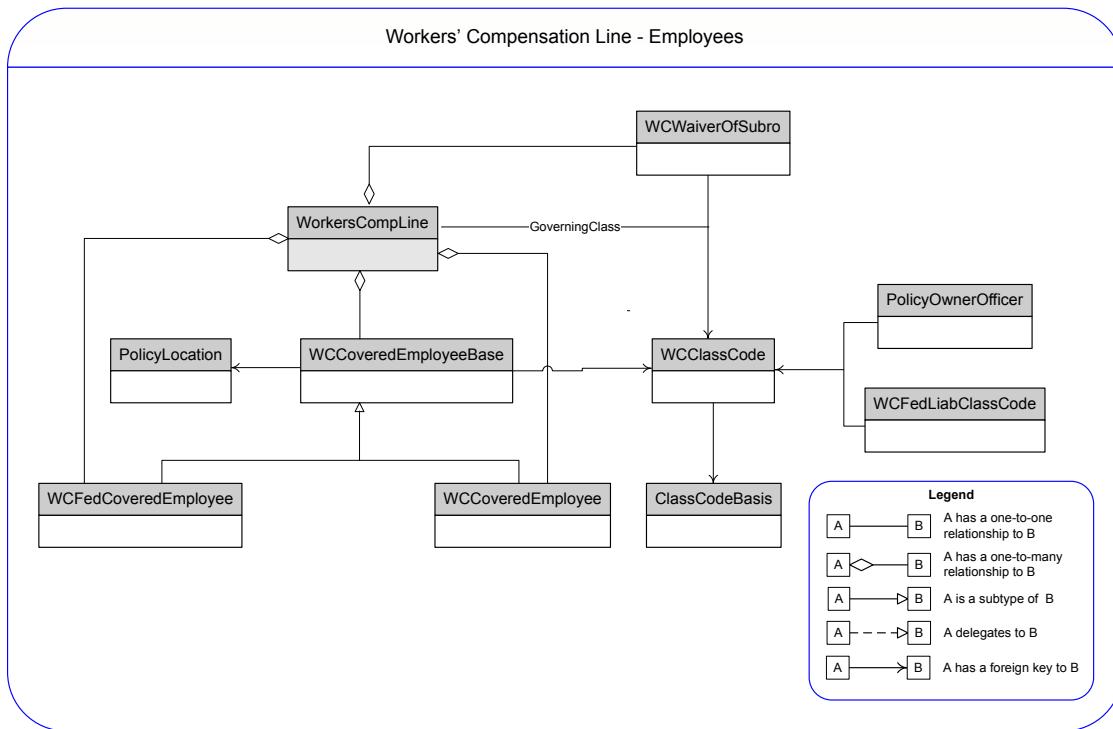
In workers' compensation, a modifier is a value used by the rating engine to adjust the policy premium or some portion of the premium. Modifiers capture information relevant to the pricing of a policy that are not necessarily tied to a specific coverable or coverage. In workers' compensation, there is 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 393.

Employee entities in workers' compensation

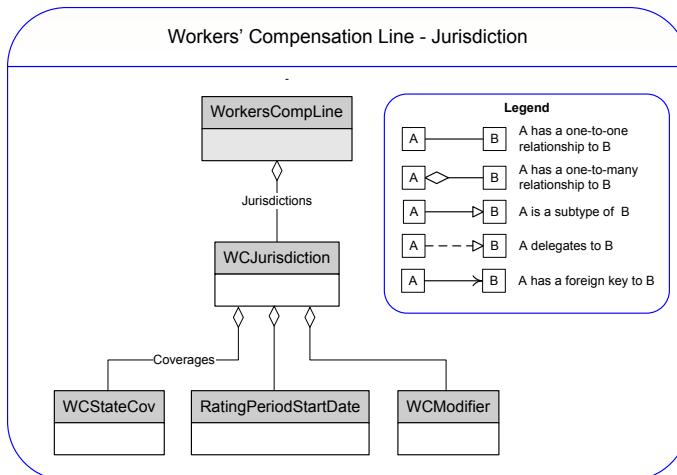
The following diagram shows entities related to employees in workers' compensation. PolicyCenter represents each employee class by a `WCCoveredEmployeeBase`, a `WCCoveredEmployee`, or a `WCFedCoveredEmployee` entity. The `WCCoveredEmployee` subtype has an array key that allows you to get to `WCCovEmpCost`. 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*.

Jurisdiction entity in worker's compensation

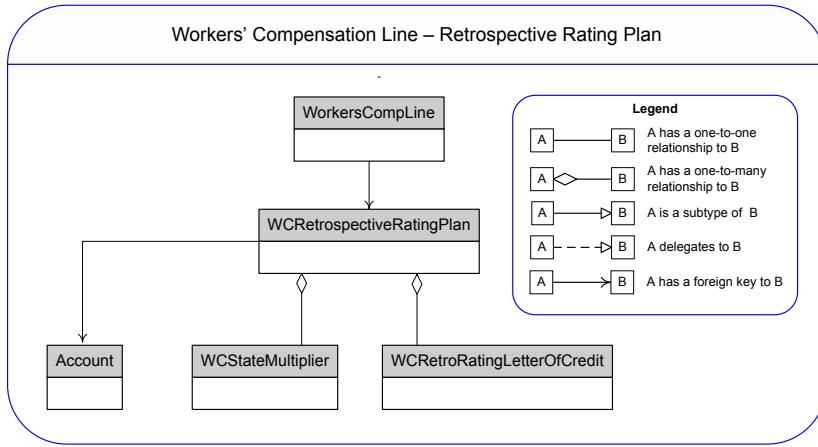
In workers' compensation, the **WCJurisdiction** entity maps to a covered jurisdiction. The **State** property is a type key to the jurisdiction. Other properties store the effective and expiration dates and, if the period is sliced, the slice date. The **WCJurisdiction** entity allows you to access 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 entity in workers' compensation

In workers' compensation, 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*.

Part 5

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.

See also

- “Policy transactions” on page 83

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 83.

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 buy 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 insurer. The insurer 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 insurer cancels the original policy.
- **Spin-off** – A son moves out of the house, and takes a car covered on his parents' policy. The insurer 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 insurer 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.

Earned premium

The **Earned Premium** is the portion of premium that applies to the expired part of the policy period. In other words, the amount of premium that has been earned as of the current date. For reporting policies, prior to final audit, the calculation includes the earned-but-unreported (EBUR) amount. For package policies, earned premium is shown for each line of business. You cannot edit the value of this field.

Click **Calculate Earned Amount as of different date** to see the earned premium on a different date. This does not affect the calculation on the **Summary** screen.

See also

- *Configuration Guide*.

Loss ratio

The **Loss Ratio** represents the total loss incurred for claims divided by the current earned premium. The claim system provides the claims amount. If enabled, the built-in integration with ClaimCenter provides the total loss incurred. If you are not integrated with a claim system, the loss ratio is always 0.

The loss ratio fields are not automatically updated each time you display the **Summary** screen. Click **Recalculate Loss Ratio** to update these fields.

The equation for loss ratio, expressed as a percentage, is:

```
Loss Ratio = 100 * Total Loss Incurred / Earned Premium
```

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.

Info bar in policy file

In the policy file, use the Info bar at the top of the screen 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 in policy file

When viewing the policy file, 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 in policy file

The **Policy Info** screen in 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 share certain steps, such as **Policy Info**, **Forms**, and **Payment**.

Tools menu in policy file

The **Tools** menu 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.
- **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 the *Configuration Guide*.

Policy Summary in policy file

General use

The screen summarizes information about a policy, providing useful information for underwriters and other people who make decisions about policies. For example, it provides all the information you need when deciding to renew a policy.

In this screen, you can:

- Change, cancel, or renew a policy
- Review the financials of this policy
- Check the loss ratio on this account
- Review or add notes about the account
- Check latest policy terms, transactions and claims

See also

- *Configuration Guide*

How to access

This summary screen is available:

- In the **Policy** tab
- From any screen that contains the reference to a policy, for example **Account Holder Summary**, or **Account Summary**
- Through search results

Panels in this screen

Details

Displays information about the policy, so that you can make sure you are looking at the right one. You can:

- Go to the **Account Holder Summary** window by clicking the link in the **Primary Named Insured** field.
- Depending on the status of this policy, you can perform different actions in the **New Transaction** menu. For example, you can change, cancel, or renew the policy.

Term Financials

Displays financial information related to this policy term only, such as total premium for this term. You can check the loss ratio for this policy. To get the latest information, click **Recalculate Loss Ratio**.

In final audit of a workers compensation policy, there is more info in this panel about what the earned premium will be. Seeing the estimated premium, the underwriter can better determine the value of this policy.

If the policy is a commercial package there is no earned premium in each policy, but earned premium exists in the package.

Current Activities

Displays the last five activities for this policy. It displays open activities first, starting with the highest priority and latest. You can open an activity by clicking its subject. If there are more than five activities, you can view them all by clicking **View more**.

Pending Policy Transactions

Displays the last five policy transactions which are still open. You can:

- Open a transaction by clicking its number.
- If there are more than five policy transactions, you can view them all by clicking **View more**.

Claims

If PolicyCenter is integrated with a claim system, displays the status of the latest five open claims for this account holder. (For example, if integrated with Guidewire ClaimCenter, PolicyCenter will get the information from ClaimCenter.) You can use this panel to answer questions or decide what to do about a request from your customer. You can view more details about the claim by clicking its number. This action redirects you to your claim system.

If there are more than five claims, you can view them all by clicking **View more**.

Completed Policy Transactions

Displays the last five policy transactions which are completed. You can:

- Open a transaction by clicking its number.
- If there are more than five policy transactions, you can view them all by clicking **View more**.

Account

Displays information about the account that holds this policy. You can:

- Go to the account page by clicking the account name
- See how many in-force policies this account has
- See how many open claims this account has

Billing

If PolicyCenter is integrated with a billing system, displays a summary of billing information—how much they owe, and how much they have paid. You can use this to let the customer know when their next invoice is due, or if you received their last payment. If integrated with Guidewire BillingCenter, PolicyCenter will get the information from BillingCenter.

Contacts

Displays the contacts for this policy. You can open a contact by clicking their name. You can also see their roles in the policy, so that you know who to call about a claim, or to complete an activity. If there are more than three contacts, you can view them all by clicking **View more**.

Producer

Displays the names and codes of producers related to this policy.

If there are more than five producers, you can view them all by clicking **View more**.

Notes

Displays three latest notes. You can review past notes and add new ones by clicking **New Note**. If there are more than three notes, you can view them by clicking **View more**.

Working with policies

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

Copy data from one policy to another

About this task

You can copy data from other policy periods or other policy transactions on the current policy. To copy policy data, you must have the **Copy policy data** permission. The code for this permission is `copypolicydata`.

Procedure

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** to select data to copy from 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.

6. Select data. 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**

7. Click the **Notes** tab to copy all or selected notes.

Split an existing policy into two policies

About this task

These instructions are for *splitting* a policy. The steps for *spinning-off* a policy are similar but create a single submission instead of two.

You must have the **Split or Spin 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 `splitpolicy`.

Procedure

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.

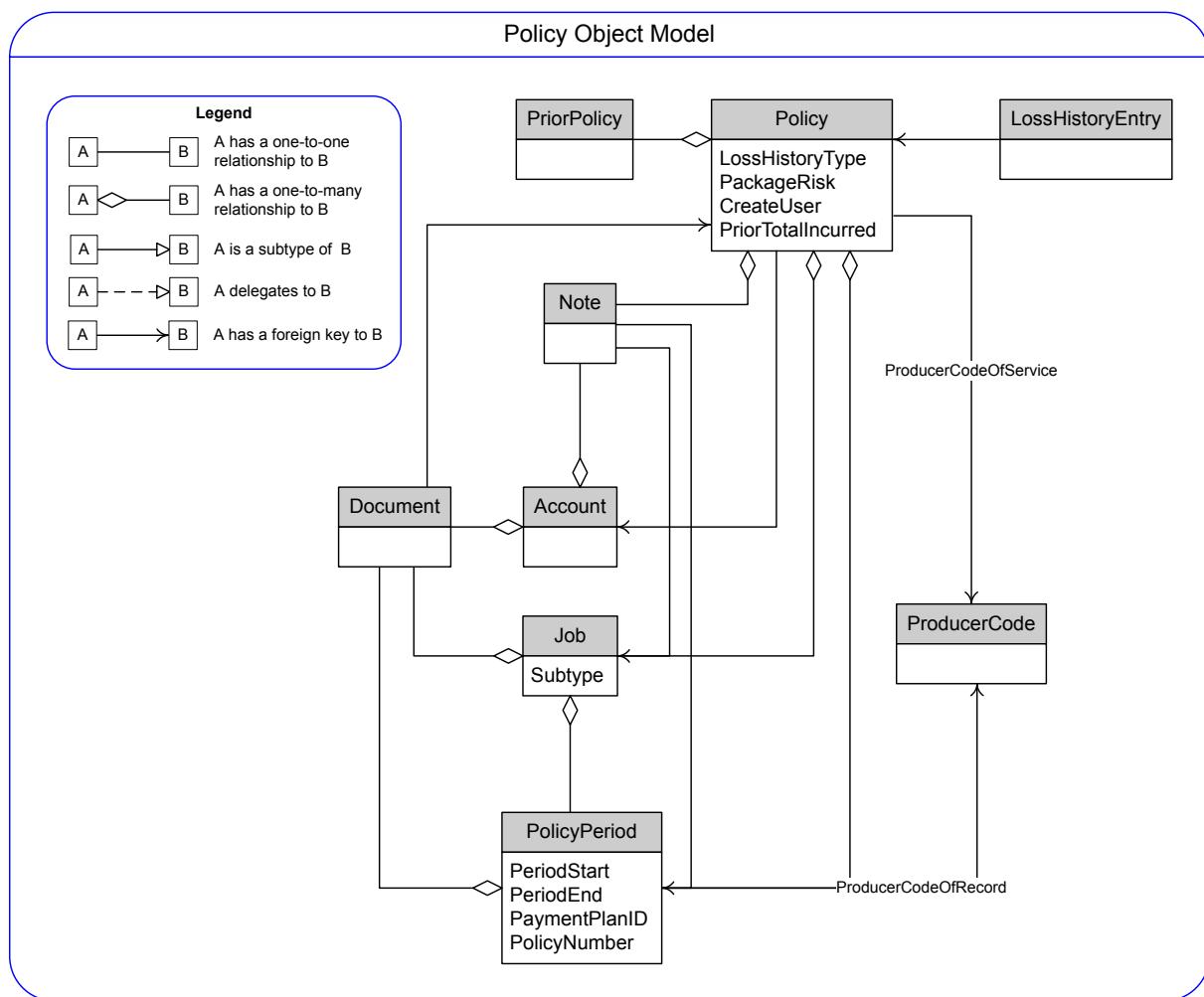
Core entities associated with policies

The entities at the core of the PolicyCenter data model are: **Policy**, **PolicyPeriod**, **Job**, **PolicyTerm**, and **PolicyLine**. The following topics provide descriptions of these and other entities.

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 an insurer 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**
- **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.

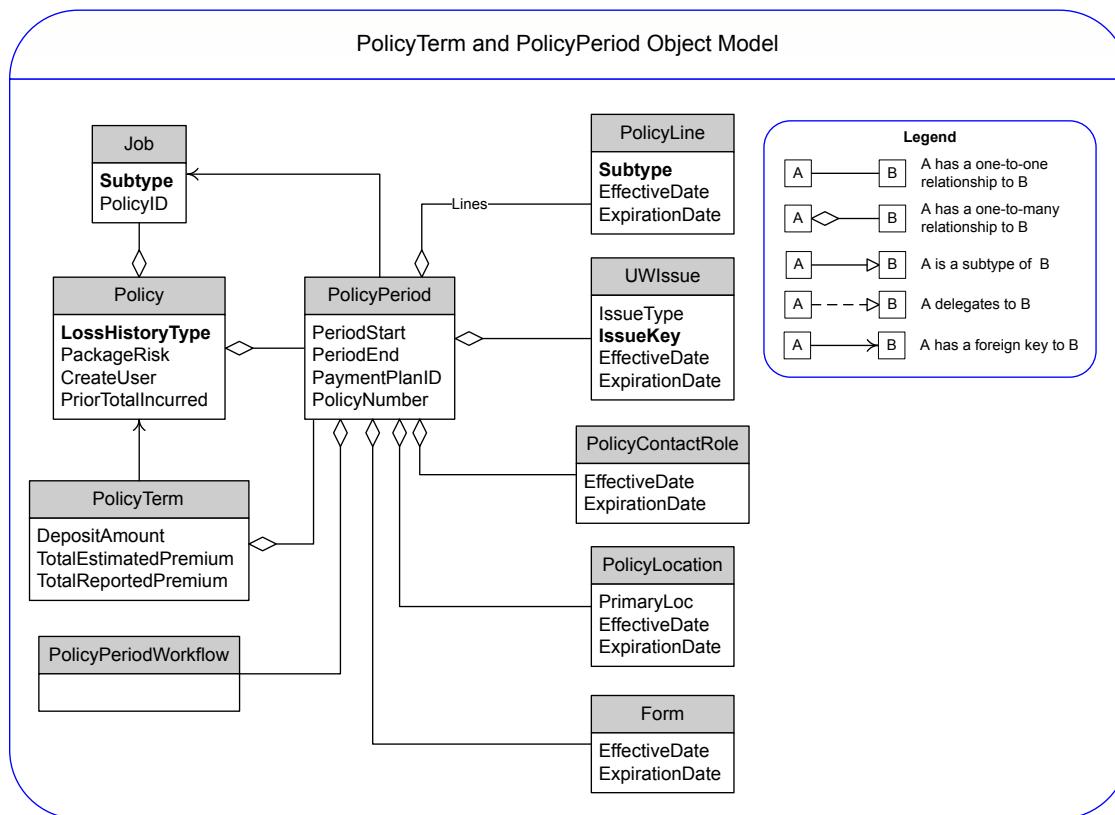
The PriorPolicy entity contains information about the prior policy.

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 entities

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 a policy term, or one contractual period for the 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`, `Cancelling`, `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 453

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

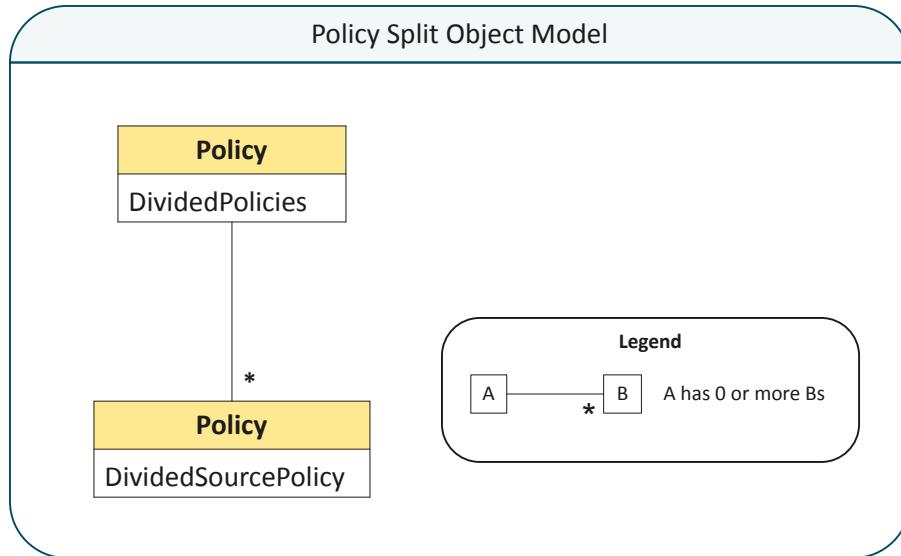
- *Product Model Guide*

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 `DividedPolicies` array on the source policy provides access to policies split or spun-off from the source policy. The `DividedSourcePolicy` 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.

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.

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.

Account security

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 706 for information on producer codes security.

Related accounts

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

- “Search for accounts with a shared contact” on page 319
- “Configuring shared contact search criteria” on page 325

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 policy transaction” on page 133

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

- “Merge accounts” on page 317

Service tier in accounts

Service tiers enable an insurer to provide special handling or value-added services for certain customers, typically high-value customers.

In PolicyCenter, the account includes a **Service Tier** field. You can view this field by navigating to the **Account Summary** screen, click **Edit** on the **Details** panel to display the **Edit Account** screen. You can set this field to **Platinum**, **Gold**, or **Silver**, in decreasing order of service. You can configure these levels of service. In the base configuration, the service tier has no effect inside of PolicyCenter.

In the base configuration, you set the service tier in PolicyCenter. If you are integrated with ClaimCenter or BillingCenter, PolicyCenter propagates the service tier on an account to:

- BillingCenter – The service tier on the BillingCenter account. Every time a submission is bound and issued in PolicyCenter, PolicyCenter sends the service tier on the account to the BillingCenter account.
- ClaimCenter – The service tier on the ClaimCenter policy. When an agent files a claim in ClaimCenter, ClaimCenter retrieves policy and account information from PolicyCenter. This information includes the service tier on the account.

See also

- *Integration Guide*

Account status

On the **Account Summary** screen, the **Status** field 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** – Changes to active when a draft submission is started for the account, a bound policy is transferred to the account, or a canceled policy is rewritten to the account.
- **Withdrawn** – Indicates that the insurer has withdrawn this account from consideration for business.

Account screens

The account menu links in the left sidebar provide supporting information for the account. This menu is context-sensitive.

Account Summary

This screen summarizes information about an account, providing useful information for underwriters and other people who make decisions about policies. This screen can also help you answer questions from agents or producers.

Account Holder Summary

This screen summarizes information about an account holder, providing useful information for customer service representatives (CSRs). The screen displays data across all accounts for which the contact is the account holder.

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.

You can enter contact information on the account and access it when creating a submission or modifying a policy.

Account File Locations

You can create locations on the account. Policy transactions can use these account locations. For example, you can access account location information when creating a submission or modifying a policy.

Account File Participants

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

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
- Policy transaction type; submission or renewal, for example
- Line of business

Submission Manager

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

The **Underwriting Files** screen displays a list of underwriting files on the account. Underwriting files are groups of policies. Underwriting files enable you to view risk information for a group of policies. Underwriting files may group policies that require processing as a group. These policies may require information from one another during the processing of their transactions. For example, the quote for the renewal of the businessowners policy requires information from the workers' compensation policy. You can simplify processing by having the two policies grouped into an underwriting file.

In the base configuration, submissions are in one group and renewals in another group. You can configure this behavior. An underwriting file corresponds to the **JobGroup** entity.

Each underwriting file group has the following tabs: **Submissions** or **Renewals**, **Risk Analysis**, and **Activities**.

Account File Related Accounts

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

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

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

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

In PolicyCenter, the **Account→Billing** screen displays account fields maintained by the billing system. For each account, you can view this page by clicking **Billing** in the left sidebar.

Account File History

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 the *Configuration Guide*.

Account Summary

General use

This screen summarizes information about an account, providing useful information for underwriters and other people who make decisions about policies. This screen can also help you answer questions from agents or producers.

- Answer customer questions about this account
- Edit a contact
- Check the loss ratio on this account
- Review or add notes about the account
- Check latest policy terms, transactions and claims

See also

- *Configuration Guide*

How to access

This summary screen is available:

- In the **Account** tab
- From any screen that contains the reference to an account, for example **Account Holder Summary**, **Policy Summary**, or **My Accounts**
- Through search results

Panels in this screen

Details

Displays information about the account contact, so that you can make sure you are looking at the right account. You can update account details by clicking **Edit**.

Current Activities

Displays the last five activities for this account and on policy transactions in the account. It displays open activities first, starting with the highest priority and latest. You can open an activity by clicking its subject. If there are more than five activities, you can view them all by clicking **View more**.

Policy Terms

Displays five policy terms, starting with future and current ones. You can:

- View the premiums and effective dates to make decisions about next steps with the account.
- View the loss ratio for this account to decide if adding more terms would be a risk. To view the latest information, click **Recalculate Loss Ratio**.
- Open a policy by clicking its number.
- If there are more than five policy terms, you can view them all by clicking **View more**.

Open Policy Transactions

Displays the last five policy transactions which are still open. You can:

- Open a policy by clicking its number.
- Open a transaction by clicking its number.
- If there are more than five policy transactions,, you can view them all by clicking **View more**.

Claims

If PolicyCenter is integrated with a claim system, displays the status of the latest five open claims for this account holder. (For example, if integrated with Guidewire ClaimCenter, PolicyCenter will get the information from ClaimCenter.) You can use this panel to answer questions or decide what to do about a request from your customer. You can view more details about the claim by clicking its number. This action redirects you to your claim system.

If there are more than five claims, you can view them all by clicking **View more**.

Overview

Displays information such as how long this account has been in the system.

If PolicyCenter is integrated with a billing system, displays high-level information about the history of this account holder, such as:

- Premium on this account in the last 3 years, including losses and loss ratio. To get the latest figure, click **Recalculate**
- Non-pay cancels in the last 12 months
- Delinquencies in the last 12 months

If integrated with Guidewire BillingCenter, PolicyCenter will get the information from BillingCenter.

You can use this information to decide about the value of the account, for example if it is a long-standing trustworthy customer, or somebody who is risky in terms of new business.

Billing

If PolicyCenter is integrated with a billing system, displays a summary of billing information—how much they owe, and how much they have paid. You can use this to let the customer know when their next invoice is due, or if you received their last payment. If integrated with Guidewire BillingCenter, PolicyCenter will get the information from BillingCenter.

Contacts

Displays the contacts for this account. You can open a contact by clicking their name. You can also see their roles in the account, so that you know who to call about a claim, or to complete an activity. If there are more than three contacts, you can view them all by clicking **View more**.

Producers

Displays the names and codes of producers related to this account.

If there are more than five producers, you can view them all by clicking **View more**.

Locations

Displays locations connected with this account. You can check an address here, before you visit the location. You can open a location by clicking its name.

If there are more than five locations, you can view them all by clicking **View more**.

Related Accounts

Displays other accounts connected with the current one. You can open an account by clicking its name. If you do not have any related accounts defined, this panel displays the accounts that the system matched automatically.

If there are more than five related accounts, you can view them all by clicking **View more**.

Notes

Displays three latest notes. You can review past notes and add new ones by clicking **New Note**. If there are more than three notes, you can view them by clicking **View more**.

Account Holder Summary

General use

This screen summarizes information about an account holder, providing useful information for customer service representatives (CSRs). The screen displays data across all accounts for which the contact is the account holder.

You can use this screen to:

- Answer customer questions about this account holder
- Start a new submission
- Edit a contact
- File a claim
- Change a policy
- Review and add notes about the account holder

How to access

This summary screen is available in the **Contact** tab, but only if the contact is an account holder.

You can access this screen:

- Through search results
- From any screen that contains the reference to an account holder, for example **Account Summary**, **Policy Summary**, or the various **Contact** screens in PolicyCenter
- From any application that uses a custom URL which points to a contact (entry point), for example from the application you use to answer customer calls

Panels in this screen

Details

Displays general information about this contact, so that you can make sure you are looking at the right contact.

You can update contact details by clicking **Edit**.

Policies

Displays the last five policies for this account holder. You can start work on these policies from here. For example, you can begin a cancellation. If PolicyCenter is integrated with a claim system, you can start a claim. If integrated with Guidewire ClaimCenter, PolicyCenter will get the information from ClaimCenter.

Open Policy Transactions

Displays the last five policy transactions which are still open. You can use this panel to answer questions about ongoing policy transactions. You can start a new submission by clicking **New Submission**.

Claims

If PolicyCenter is integrated with a claim system, displays the status of the latest five open claims for this account holder. (For example, if integrated with Guidewire ClaimCenter, PolicyCenter will get the information from ClaimCenter.) You can use this panel to answer questions or decide what to do about a request from your customer. You can view more details about the claim by clicking its number. This action redirects you to your claim system.

If there are more than five claims, you can view them all by clicking **View more**.

Overview

Displays information such as:

- How long this account holder has been your customer
- How much premium this account holder pays

If PolicyCenter is integrated with a billing and a claim system, this screen can display information such as:

- Non-pay Cancels in the last 12 months
- Delinquencies in the last 12 months
- Losses that this account holder has caused you in the last 3 years

You can use this information to decide about the value of the account, for example if it is a long-standing trustworthy customer, or somebody who keeps costing you money and is quite risky in terms of new business.

Billing

If PolicyCenter is integrated with a billing system, displays a summary of billing information—how much they owe, and how much they have paid. You can use this to let the customer know when their next invoice is due, or if you received their last payment. If integrated with Guidewire BillingCenter, PolicyCenter will get the information from BillingCenter.

Notes

Displays three latest notes. You can review past notes and add new ones by clicking **New Note**. If there are more than three notes, you can view them by clicking **View more**.

Billing screen for accounts

In PolicyCenter, the **Account→Billing** screen 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 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 enables 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. If you are in a multicurrency system, the link jumps to the primary affiliated account in BillingCenter.

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 765

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 includes topics that provide step-by-step instructions for accessing and working with accounts.

Search for an account

Procedure

1. Select the path to search for an account.

To search for an account from Path

The Account tab	Click the account if it is visible or enter the account number in the Acct# field.
The Search tab	Select Accounts .

2. For the search path, enter your search criteria and click **Search**.

Account 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	City and State
ZIP Code	PostalCode

Minimum search criteria is defined in the `hasMinimalCriteria` method in the `gw.account.AccountSearchCriteria` class.

Create an account

Procedure

1. Select the path to create an account.

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.

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.

2. 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.
3. Enter the required information and select **Update**.

The **Account Summary** screen appears, summarizing your information. The account's status is Pending, until you associate a submission with it.

4. Select an option to modify the account.
 - Edit the account
 - Change the account holder to a new person, company, or new contact from the address book
 - Add locations, account roles, notes, or documents
 - Create a submission

Track your accounts

About this task

You can track your accounts by viewing a summary of your pending accounts containing the account number, name, status, and address and filtering on the records.

Procedure

1. Select **My Accounts** from the **Desktop** tab.
2. Refine the search by filtering.
The default filters include:
 - All Pending
 - Created in Past 7 Days
 - All
3. Select an account number's link to navigate to the **Account Summary** screen.

Move a policy from one account to another

About this task

This topic describes how to move a policy from a source account to the target account.

Procedure

1. Navigate to the target account.
2. From the **Actions** menu select **Move Policies to this Account**.

The menu item appears if you have the **Move policies** permission for the target account. The code for this permission is `accountmovepolicies`.

Use the **Related to account number** check box to restrict the search to related accounts only.

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.

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 “Moving policies in the Account plugin” on page 322 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.

Rewrite policies from one account to an existing account

About this task

You can rewrite one or more policies from a source account to an existing target account.

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.

Procedure

1. Navigate to the target account.
2. From the **Actions** menu select **Rewrite Policies to this Account**.

Then menu item appears if you have the **Rewrite policies to account** permission for the target account. The code for this permission is `accountrewritepolicies`.

Use the **Related to account number** check box to restrict the search to related accounts only.

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.

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.
Product	Displays the product name of the policy.
Status	Displays the policy status as of the current date.
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.

Merge accounts

About this task

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`.

Procedure

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

- “Merging accounts” on page 306

Add an account relationship

About this task

Note: You must have the **View account file related accounts** permission to view the **Related Accounts** screen. The code for this permission is `accountrelations`.

Procedure

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.

Modify an account relationship

Procedure

1. Click 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.

2. Click **Update** to modify the relationship.

PolicyCenter validates the account.

Remove an account relationship

About this task

Note: You must have the **Edit account** permission to view the **Remove** button. The code for this permission is `editaccountsummary`.

Procedure

1. Select one or more rows on the **Related Accounts** screen.
PolicyCenter enables the **Remove** button.
2. Click the **Remove** button to remove the account relationship.

Search for accounts with a shared contact

About this task

See also

- “Configuring shared contact search criteria” on page 325

Procedure

1. Navigate to the **Related Accounts** screen.
2. Click **Search for Accounts with a common account holder or named insured**.

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.

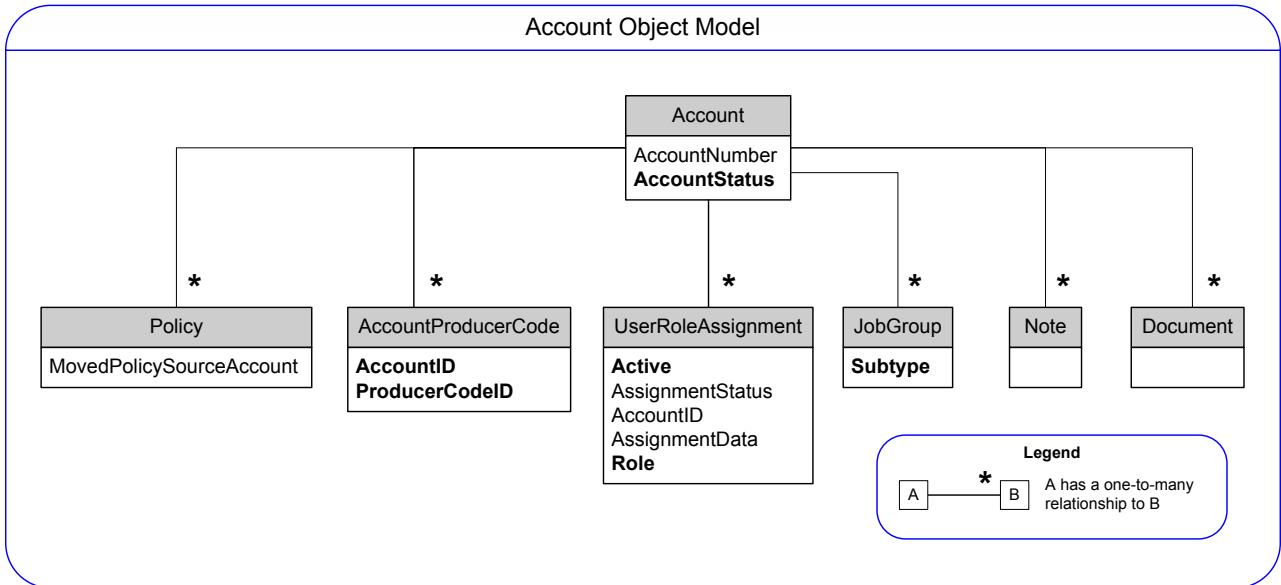
Configuring accounts

This topic describes some of the basics of account configuration.

Account object model

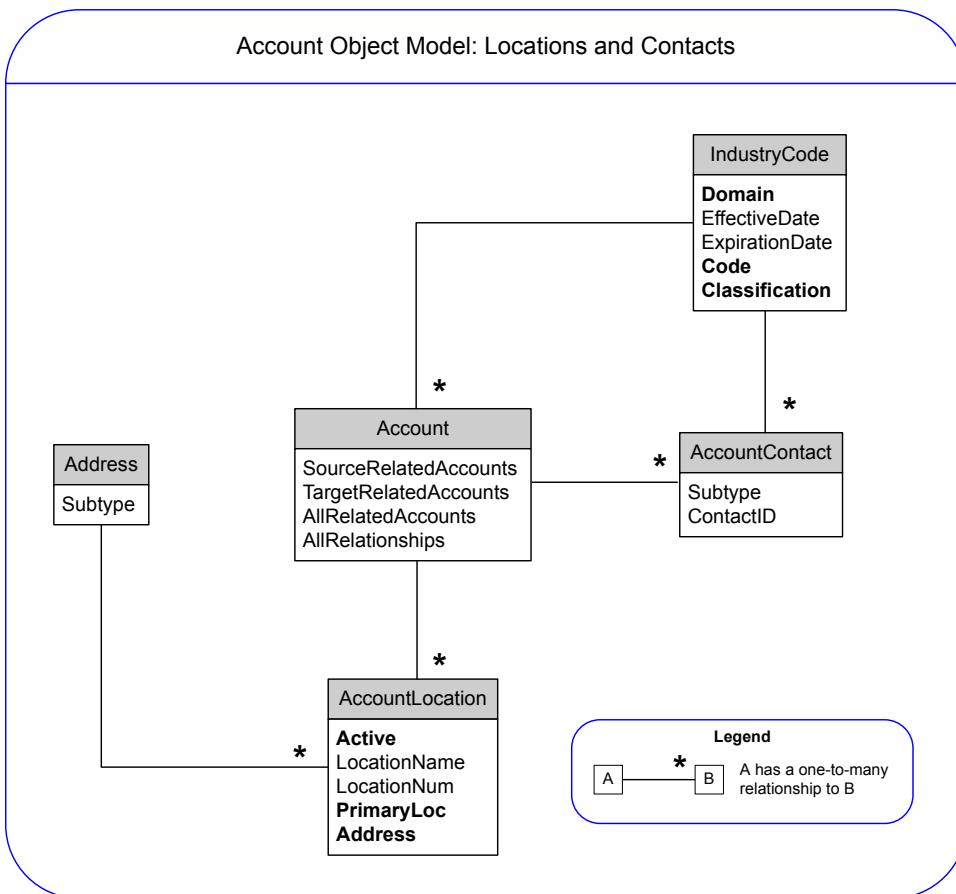
The **Account** object model helps you to better understand the entity relationships of accounts.

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 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 the *Integration Guide*.

Withdraw accounts

Account Withdraw Evaluation batch processing changes the status of accounts to withdrawn.

The Account Withdraw Evaluation work queue marks the account status as withdrawn (**Withdrawn**) if:

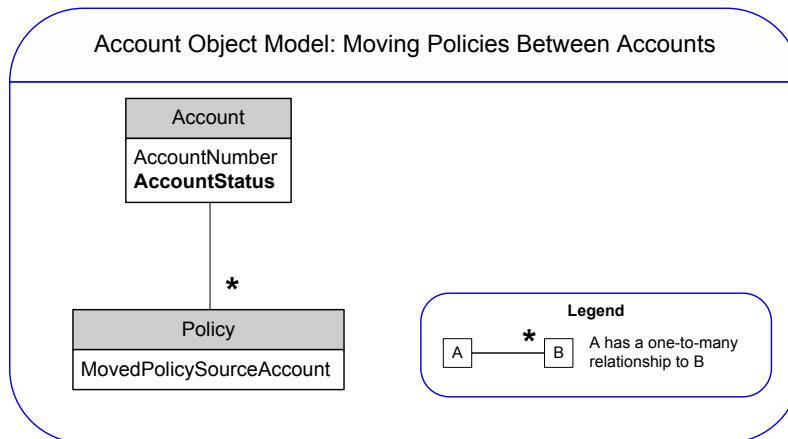
- There are no policies associated with the account.
- The `Account.CreateTime` or `Account.OriginationDate` is older than a configurable number of months in the past. The `AccountsWithdrawnAfterMonths` parameter in `config.xml` specifies the number of months. In the base configuration, this parameter is set to 37 months.
- There are no open activities associated with the account.

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.

Moving 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.

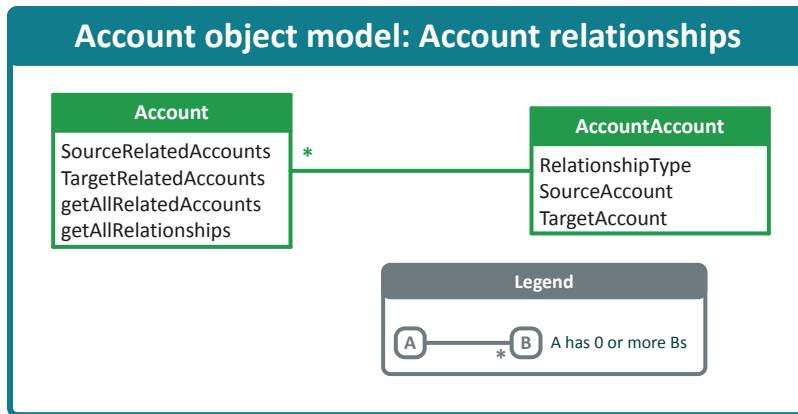
Configuring account relationships

This includes topics that describe how to configure account relationships.

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 323.

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 `getAllRelatedAccounts` method returns 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 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 the *Rules Guide*.

The following table describes rule sets in the default application related to account relationships.

Rule	Description
<code>Validation→Account Validation Rules→Related Accounts</code>	Invoked when an <code>AccountAccount</code> entity is added or updated. Performs the following checks: <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. If necessary, you can modify or add additional validations.

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

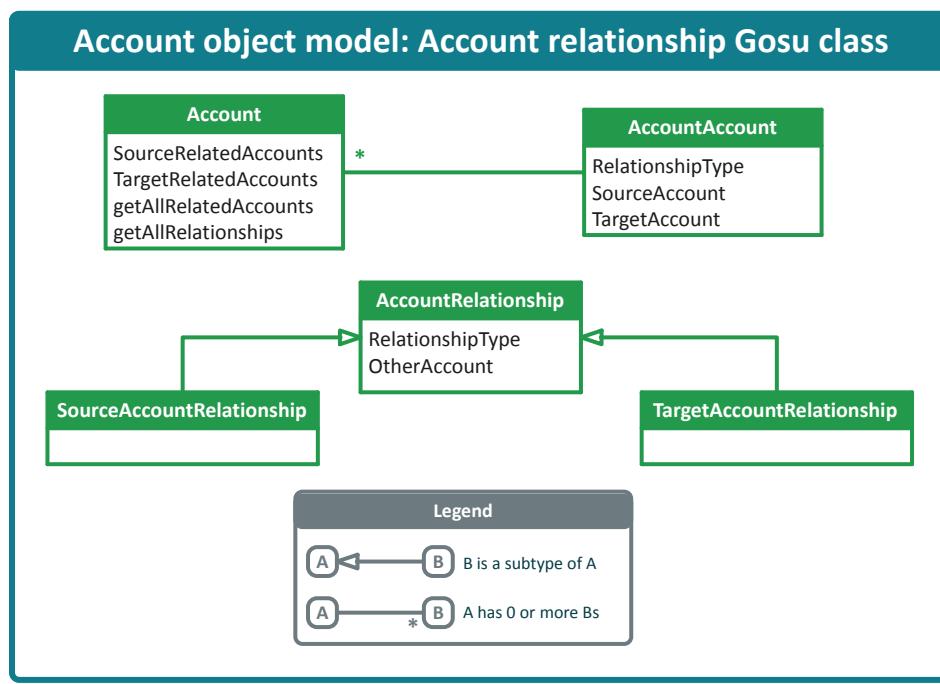
Getting relationships

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.getAllRelationships method in AccountBaseEnhancement returns an array of AccountRelationships for all relationships of an account.

```
Account.getAllRelationships : 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.

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

- *Configuration Guide*
- *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 revised.

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 revised. In the default configuration, the location name, location number, and phone number are not revised. 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 revised 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 “Add a new location” on page 331 for details.

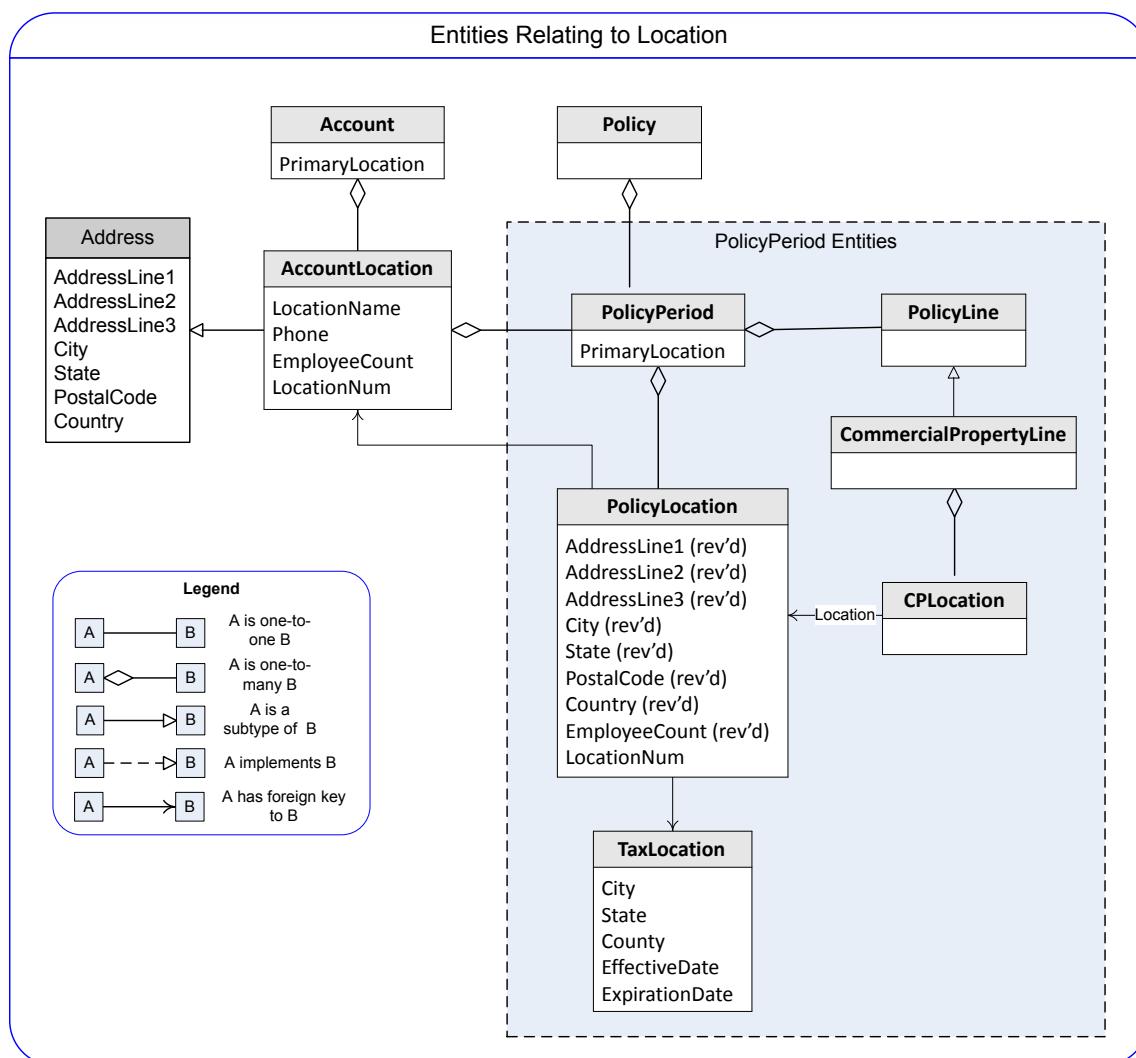
Hiding location information on accounts

The default configuration of PolicyCenter has locations viewable and editable at the account level. However, some insurers 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 revised 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 has 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.

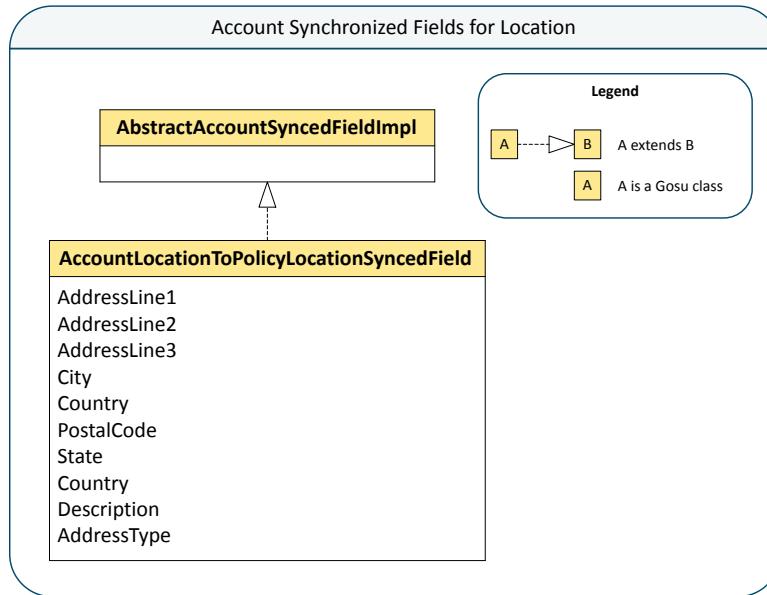
Some of the subtypes of `PolicyLine` entity have associated location types. The subtypes have a foreign key, `Location`, that points to the `PolicyLocation`. The object model diagram shows the `CommercialPropertyLine` subtype.

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>

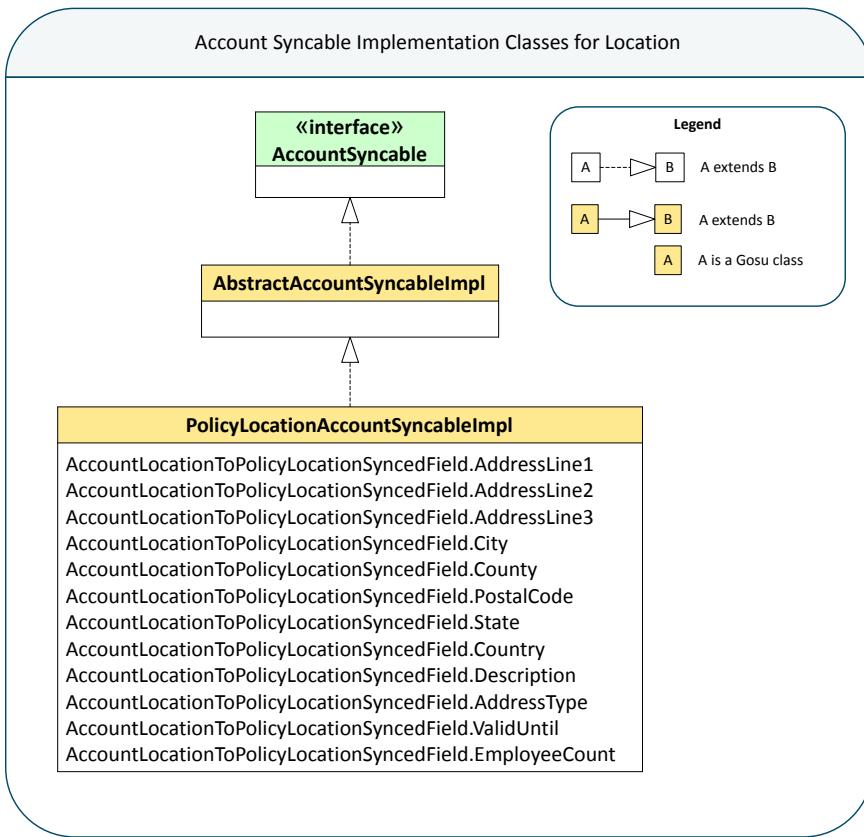
In the default configuration, account location and policy locations are numbered separately. Because of this, both `AccountLocation` and `PolicyLocation` have their own `LocationNum` field which is not revisioned. For more information about location numbering, see “Location numbering” on page 333.

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 314.

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 included topics assume that you already know how to access an account.

Add a new location

Before you begin

Before adding a new location, you must already have selected or created an account.

Procedure

1. Navigate to the **Account Summary** screen for an account.
2. On the sidebar, click **Locations**. The **Account File Locations** screen appears.
3. Click **Add New Location**. The **Location Information** screen appears.
4. Enter your data and click **Update**. The **Account File Locations** screen appears.

Edit a location

About this task

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.

Note: When editing a location, you cannot change the following fields: **Non-Specific Location**, **State**, and **Country**.

You must first find the account of the location you want to modify.

Procedure

1. In an account, navigate to the **Account File Locations** screen.
2. Under the **Account File Locations**, select the location number link.
3. The **Location Information** screen opens in edit mode. Make your changes and click **Update**.

Make a location primary

About this task

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.

Procedure

1. In an account, navigate to the **Account File Locations** screen.
2. On the **Account File Locations** header, select the check box of the location you want to make primary.
3. Click **Set as Primary**.

Make a location active or inactive

About this task

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.

Procedure

1. In an account, navigate to the **Account File Locations** screen.
2. On the **Account File Locations** header, select the check box of the location you want to change.
3. 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 included topics assume that you have already created at least one account and one policy.

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.

View location changes

About this task

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**.

Procedure

1. 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!**
2. 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.

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 makes 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.

Activities overview

In PolicyCenter, there can be many tasks that need to occur before a policy transaction can finish. For example, before issuing a commercial 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 insurer 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

- “Create an activity” on page 337 for information on how to manually create an activity.
- “Assign an activity” on page 337 for information on how to assign one or more activities.
- The *Rules Guide* for information on how PolicyCenter assigns an activity.
- 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

- *Configuration Guide*

Working with activities

This includes topics that explain how to work with activities.

Create an activity

About this task

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.

Procedure

1. 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.
2. Select **New Activity** from the **Actions** menu. Select the category (**Reminder** in this example) and the type of activity (**VerifyCoverage**).
3. Enter the required information. You can either select a person to handle the assignment or have PolicyCenter assign it for you.
4. Optionally, you can add a **New Note** at the same time.
5. 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.

Assign an activity

About this task

You can reassign an activity that you own.

Procedure

1. Select the path to reassign an activity.
 - Navigate to the activity and click **Assign**.
 - Navigate to a list containing the activity.
2. From the list path, select the check box to the left of one or more activities.
3. Click **Assign**.
4. Either path displays the **Assign Activities** screen where you can assign the activity through assignment, or specify a user, group, or queue.

Complete or skip an activity

About this task

You can update, skip, or complete an activity.

Procedure

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**.

Select an activity from a queue

About this task

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.

The user must have the **Get next activity from queue** (actquenext) permission. In the base configuration, the underwriter role has this permission.

Procedure

1. Navigate to **My Queues**, select from the drop-down menu, and you can see activities in the queue.
2. 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.
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.

Field	Description
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"> • Correspondence • Interview • New mail • Reminder • Request • Response • 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 or Assignment Review activity. General activities are closed by being completed or skipped. Assignment Review activities are added to a supervisor's Pending Assignment queue.
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.

Create a new activity pattern

Procedure

1. Click **New Activity Pattern** on the **Activity Patterns** screen.
2. Enter the data as described in “Activity pattern components” on page 338.

3. To save, click **Update**.

Modify an existing activity pattern

Procedure

1. Select the desired pattern.
2. Enter the data as described in “Activity pattern components” on page 338.
3. To save, click **Update**.

Using Gosu to edit activity patterns

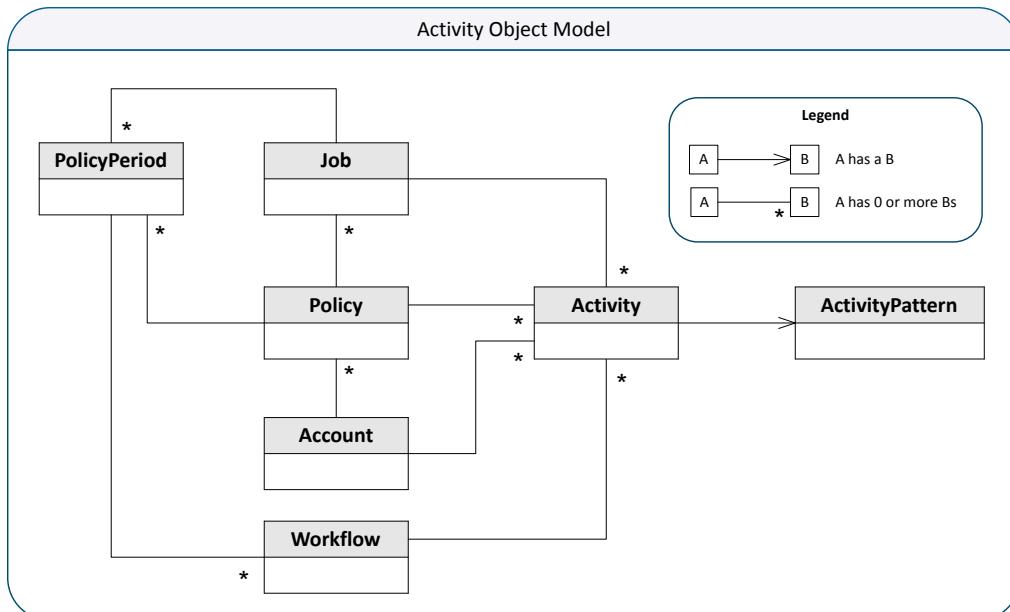
You can also create activity patterns by using Gosu in Studio.

See also

- *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 338 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.

Entity	Description
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

- *System Administration Guide*

Notes

Notes can be used by PolicyCenter end users to capture information about an account or policy that does not easily fit anywhere else.

You can use the Notes feature to:

- Create general notes without a note template.
- Create notes with a note template for specific note types.
- Add additional security with ACLs.
- Edit and delete notes, if you have the proper permission.
- Search for notes with a wide variety of filters.
- Create a note with rules or in workflows.
- Create new note templates.

Differences between notes and documents

Notes and documents have distinct functions in PolicyCenter, and the application handles them differently. The following table highlights the main differences.

Characteristics of Notes	Characteristics of Documents
Written in plain text.	Can have many different MIME types, such as PDF, Word, or Excel.
Created by a user or through Gosu.	Created by a user, or through Gosu, or in an external document management system.
Stored only in the PolicyCenter database.	Stored either in the PolicyCenter database or in a document management system.

See also

- “Document management” on page 677

Working with notes

The primary screens and worksheets related to notes are:

- **Notes** – This screen has a search area in its upper part with fields that enable you to search by text, author, topic, and date range. The main screen shows the results of the last search in its lower section.
- **New Note** – This worksheet is where you create notes. You can optionally use a template. You can also search for note templates. To access this worksheet, click **Actions**→**New**→**Note**.
- **Activity Detail** – This worksheet is where you create notes that are related to an activity. For example, you can navigate to **Desktop** tab **Activities**. If you then click the **Subject** of an activity assigned to you, **Activity Detail** worksheet of that activity opens below the list of activities. The worksheet has a **New Note** section.

Searching for notes

To search for notes, open an account or a policy and click **Notes** in the sidebar to open the **Notes** screen. Use the search section at the top of the **Notes** screen to search for notes.

You can use the following filters and search fields:

- **Text Search** – Search for a word or text string in the subject or body of the note.
- **Author** – The user who wrote the note. PolicyCenter attaches the name of the user to a note when it is created. You cannot change the author of a note.
- **Related To** – A note created in a policy, submission, or activity is related to that entity. A note can be related to just one entity.
- **Topic** – A typecode that classifies the note, which you choose from a drop-down list. This list is populated from the **NoteTopicType** typelist.
- **Date Range** – Enter a range of dates.
- **Sort By** – Sort the results by author, date, exposure, subject, or topic in either ascending or descending order. These values are typecodes from the **SortByRange** typelist. In the base configuration, the default sort is by date in descending order.

In the base configuration, you cannot search by **Security Level**.

Viewing notes

Use the **Notes** screen to see the most recent notes, and use the upper search section of the screen to find notes. If the note appears in a list, click it to see it.

To view the details of a note, click **Edit**. All the note's attributes display in the **Edit Note** screen.

You can configure the **Notes** screen to show more than the default information available in the base configuration.

See also

- “Configuring notes and note templates” on page 346

View notes related to an activity

About this task

A note created in an **Activity** worksheet is linked to that activity.

Procedure

1. Click **View Notes** on the **Activity Detail** screen.
A search screen opens that is similar to the **Search** pane on the **Notes** screen.
2. Use the search screen to find notes linked to the activity.

Edit a note

Before you begin

If you have the **noteedit** and **noteeditbody** permissions, you can click **Edit** for each note.

Procedure

1. Click **Edit** to start editing.
2. Click **Update** to save.

Delete a note

Before you begin

If you have the **notedelete** permission, you can click **Delete** for each note.

About this task

Note: The note is deleted immediately, so be sure you want to delete it before clicking **Delete**.

Click **Delete** to delete a note.

How to print a note

To print a note, you must have permission to view it.

You can print a note from the **Notes** screen's list of notes.

Click **Print** for any note you see in the list.

The note is converted to a PDF file. You can view the PDF file in Acrobat Reader and print from that application, or you can save the file.

Create a note

Procedure

1. In PolicyCenter, choose **Actions**→**New Note** to open a **Note** worksheet.
2. Choose values for the required attribute fields—**Topic**, **Related To**, and **Security Level**—and optionally fill in the **Subject** attribute. The **Security Level** field specifies the access control list (ACL) for the note.
3. Enter the note text. Notes must always contain some text.
4. Click **Update** after you are finished with your note.

Create a note from a note template

Before you begin

Before using a template, you must first find one.

About this task

You can use a template to create a note.

Procedure

1. In PolicyCenter, choose **Actions**→**New Note**.
2. In the **Note** worksheet, click **Use Note Template**.
3. In the **Pick Note Template** screen, optionally select template attributes to limit the search, and then click **Search**.

The search returns a list of templates matching your search criteria, or all templates if you enter no criteria.

4. Click **Select** to choose the template to use for creating the note.

After you select a template, the template's attributes and text populate the **Note** worksheet.

5. Change any information added by the template and edit other fields and body text as needed.

6. Click **Update** when you are finished.

Creating a note in an activity

To create a note associated with an activity, use the **New Note** section in the **Activity Details** worksheet.

Note security

PolicyCenter provides a set of system permissions to provide security for all notes, listed in “Permissions related to notes” on page 346. Use these permissions to define different security types for notes and assign permissions to users that relate to these ACLs.

Select the ACL to which you want the note to belong by specifying its **Security Level** when you create the note.

Permissions related to notes

The following system permissions provide security for notes.

Permission	Code	Description
Create notes	notecreate	Permission to add notes.
Delete notes	notedelete	Permission to remove notes.
Delete internal notes	delintnote	Permission to delete an internal note.
Delete sensitive note	delsensnote	Permission to delete a sensitive note.
Edit note	noteedit	Permission to edit any part of a note.
Edit note body	noteeditbody	Permission to edit the body of notes. Note: Although this permission exists, the functionality is not implemented in PolicyCenter.
Edit internal notes	editintnote	Permission to edit an internal note.
Edit sensitive note	editsensnote	Permission to edit a sensitive note.
View notes	noteview	Permission to view notes.
View sensitive note	viewsensnote	Permission to view sensitive notes.

Configuring notes and note templates

The Notes feature requires little configuration.

You cannot add and delete search filters in the `search-config.xml` file, as you can for other types of searches.

Note plugin interfaces

There are two plugin interfaces associated with notes. They do not affect the primary use of notes, which are stored in the database and do not require an external system, such as a document management system. They are related to note templates, which can be a customized method of creating notes. The plugin interfaces are:

- **INoteTemplateSource** – Retrieves note templates—**INoteTemplateDescriptor** objects—that are used to help create notes. The default implementation is `gw.plugin.note.impl.LocalNoteTemplateSource`. This Gosu

class retrieves templates from the server file system, but can also be customized to get them from a document management system.

- **INoteTemplateSerializer** – Customizes reading and writing of **INoteTemplateDescriptor** objects.

Note fields

Notes and note templates have a set of fields, also called properties. PolicyCenter uses these fields to attach the notes to various policy entities and to search for notes and note templates.

The following table describes the fields of a Note that are visible in PolicyCenter screens.

Attribute Name	Definition of Attribute	How Set	Search for Note?	Editable?
Author	Logged-in user who wrote the note	By PolicyCenter	yes	no
Body	Contents, the text of the note	By author in editor	yes - any string	yes
AuthoringDate	Date the note was originally written	By PolicyCenter	yes - and by range	no
NoteRelatedTo	Must exist and be unique	By author in editor	yes	yes
SecurityType	Value from the NoteSecurityType typelist	By author in editor	no	yes
Topic	Value from the NoteTopicType typelist	By author in editor or by template	yes	yes
Subject	Defined in the template and given to its notes	By author in editor or by template	no	yes

The author, body, date, related to, confidential, and security type are fields unique to notes and are not a part of note templates.

Security type is the **Security Level** in the user interface.

The following fields are used in note templates. The first two are applied by the note template to a note created from it.

Field Name	Definition of Field	Search for Template?	Search for Note?	Editable in Note?
Subject	The subject of the template and of notes created from it.	no	no	yes
Topic	The topic of the template and of notes created from it. A typecode of the NoteTopicType typelist.	yes	yes	yes
Type	A typecode of the NoteType typelist, such as diagram, action plan, or status report. You can add others.	yes	no	no

Creating a note template

A note template is a pair of Gosu files, a .gosu file and a .gosu.descriptor file. To access the files, open Guidewire Studio and navigate in the **Project** window to **configuration→config→resources→notetemplates**.

The easiest way to create a new template is to modify copies of two existing files for one of these templates. Then save the two files with new, matching names in the same location as the other note template files.

Note template files

A note template consists of two separate files:

- A descriptor file with a name ending in .gosu.descriptor. The file contains metadata about the template.

For example, `ActivityActionPlan.gosu.descriptor`.

- A template file with a name ending in `.gosu`. This file contains the text for the body of the note.

For example, `ActivityActionPlan.gosu`.

In the base configuration, a note descriptor file has the following fields.

Field	Description
<code>name</code>	A String value that is a unique, readable name for the template. Can be used in template search.
<code>type</code>	A String value that is the type of the note, a string that matches a typecode from the <code>NoteType</code> typelist. Can be used in template search. Base configuration values include <code>actionplan</code> , <code>diagram</code> , <code>interviewreport</code> , and <code>statusreport</code> .
<code>lob</code>	The product that the note is associated with. For example, Commercial Package or Workers' Compensation. Can be used in template search.
<code>keywords</code>	A String value, a comma-separated list of keywords that can be used to search for the template.
<code>topic</code>	The topic of the note, a String value that matches a typecode from the <code>NoteTopicType</code> typelist. Can be used in template search.
<code>subject</code>	The subject of the notes created with this template, a String value.
<code>body</code>	A String value that is the name of the Gosu file containing the body of the note. Be sure to include the <code>.gosu</code> extension.

See also

- An external system can retrieve and validate note templates. For information, see “Document management” on page 677
- *Integration Guide*

Contingencies

PolicyCenter manages work associated with creating or changing a policy through policy transactions. A policy transaction is completed when the policy is bound and issued, or when the policy transaction is withdrawn. However, the policy may still require additional work. You can use *contingencies* to manage this additional work and take actions, including policy change or cancellation transactions, if conditions are not met in a timely manner. For example, a personal auto policy provides a good-student discount which is contingent on receiving the driver's grade transcript. If the transcript is not received promptly, the policy is changed to remove the discount. After issuing a policy, commercial lines may have an underwriting period. During this period, the insurer inspects the property, evaluates the risk, requests remediation of conditions, and potentially cancels or changes the policy as a result. You can use contingencies to manage this work.

See also

- *Configuration Guide*

Contingency overview

A contingency is associated with a policy, not a policy transaction, and can exist past any specific policy transaction. If sufficient time elapses and the contingency is not resolved, the contingency triggers an action related to the policy. This action might be a change in policy terms or pricing, or a cancellation of the policy.

Each contingency has a number of fields including a title, description, and status. You can attach documents, notes, and activities to contingencies. Each contingency has an action, typically a policy change or cancellation, and an action start date. If the contingency is not resolved by the start date, PolicyCenter batch processing starts the specified action, for example starting a policy transaction.

Working with contingencies

You can create and view contingencies in policy transactions and in the policy file.

Viewing contingencies

You can view contingencies associated with any policy or policy transaction. To view contingencies, navigate to the **Risk Analysis** screen and select the **Contingencies** tab. This screen provides access to both **Pending** and **Resolved/Waived** contingencies. For each contingency, the screen displays fields that include title, status, due date, and potential action if not resolved. The screen displaying an individual contingency shows associated notes, documents, and open and completed activities.

Policies and policy transactions clearly indicate the presence of unresolved contingencies. In a policy transaction, a message alerting you to the presence of unresolved, or outstanding, contingencies appears below the title on the **Policy Info**, **Risk Analysis**, **Quote**, and **Forms** screens. The message appears whether the policy transaction is open or bound. In the policy file, this message appears on the **Summary** and **Risk Analysis** screens.

Creating contingencies

PolicyCenter initiates action for a contingency that is not resolved or waived within a configurable amount of time before the due date of the contingency. The configurable amount of time is the *action start date*. In the base configuration, the action start date triggers either a policy change or cancellation policy transaction. The policy transaction can either take effect retroactively or apply to the remainder of the of the policy period.

Although PolicyCenter initiates the policy change, users must manage the policy change as they would any other. Cancellations are processed as any other cancellation. Cancellations generally bind on their effective date unless the cancellation is withdrawn.

Examples

Change policy retroactively starts a policy change five days before the due date. The effective date of the policy change is the policy period start date.

Cancel remainder of term starts a cancellation 30 days before the due date. The effective date of the cancellation is one day after the due date.

Create a contingency

About this task

You can create a new contingency for a policy from within an open policy transaction and also from the policy file.

Procedure

1. Navigate to the **Risk Analysis**→**Contingencies** tab, and click **Add Contingency**.
2. Enter text in the **Title** and **Description** fields.
 - For title, enter a brief description enabling a user to quickly understand the type of contingency.
 - For description, enter a longer description which can include specific details of this instance. This ca include the conditions under which the contingency can be resolved.
3. From the drop-down list, select an **Action** that will result if the contingency is not resolved by the **Due Date**.

For each action the following table lists the action start date and the effective date of the policy transaction:

Action	Action start date	Effective date
Policy change		
Change policy retroactively	5 days before the due date	Policy period start date
Change policy for remainder of term	5 day before the due date	One day after the due date
Cancellation		
Cancel retroactively	30 days before the due date	Policy period start date
Cancel remainder of term	30 days before the due date	One day after the due date
Cancel / Rewrite	30 days before the due date	Policy period start date

The action start date is calculated in the `updateActionStartDate` method in `ContingencyEnhancement.gsx`.

4. In **Due Date**, enter the due date for the contingency. The due date is used to calculate the action start date. The Handle Unresolved Contingency work queue initiates action on pending contingencies with an action start date of today or a date in the past for contingencies on which action has not yet started.

Adding notes, documents, and activities to contingencies

On the **Contingency** screen, you can add notes, documents, and activities to an existing contingency, regardless of the contingency status. These notes, documents, and activities are directly related to the contingency. Through configuration, these notes, documents, and activities may be related to other entities.

See also

- “View documents for a contingency” on page 680

Changing contingency status

On the **Contingency** screen, you can change the status of a pending contingency to resolved or waived. Both **Resolved** and **Waived** status indicate that the contingency is closed. The **Closed By** and **Closed On** fields indicate by whom and when.

A contingency can be resolved or waived regardless whether it has open activities. Similarly, closing all activities related to a contingency does not resolve the contingency.

If the contingency is not closed by the due date, PolicyCenter changes the status to **Failed**.

If you close or waive a contingency and the contingency started a policy transaction, that transaction is unaffected by the change in contingency status. Therefore, if you resolve or waive a contingency that is past its action date, you, the user, need to take appropriate action. In some simple cases, the appropriate action is to close the policy transaction.

In other complex cases, the appropriate action is to continue the policy transaction even though the driving contingency has been resolved. For example, during the underwriting period for a commercial policy, on-site inspection reveals that the policy rating is not accurate. You modify the policy change to reflect these rating changes. Or, if documents have been sent to the insured or third parties, further action on the policy may be required. This action is more complex than simply withdrawing the policy transaction.

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 327.

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 357
- “Configuring the Contact tab” on page 378
- “Setting preferences” on page 42 for information about setting the number of recent contacts that PolicyCenter displays on the **Contact** tab.

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 789
- “Adding a contact from the address book” on page 363
- *Guidewire 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 355

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.

Revisioning contact information in policies

In policies, some contact data is revised. All revised 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 revised policy data. Revised 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 453

Contact revisioning when contacts are synchronized

When contacts are synchronized with the account, PolicyCenter:

- Copies contact information from the account at the start of the policy transaction.
- Copies a change to the policy contact back to the account contact throughout the steps the policy transaction.
- Copies a change to the account contact 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 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 work queue. In the default configuration, this work queue runs nightly. On the day that the change takes effect, the work queue updates the account contact. For more information, see “Contact batch process” on page 378.

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 work queue 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 “Contact revisioning in future-dated changes” on page 356.

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 366
- “Linked addresses object model” on page 372
- “Configuring linked addresses for contacts” on page 391
- *Guidewire Contact Management Guide*

Working with the Contact tab

The following topics describe the **Contact** tab functionality and implementation details.

See also

- “Centralized view of contacts on the Contact tab” on page 353
- “Configuring the Contact tab” on page 378

Create a new contact

Procedure

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.
3. If **Check for Duplicates** appears, click this button to verify that the contact does not already exist in the contact management system.

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 `ContactSystemPlugin` interface 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`.

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 793

Search for a contact

Procedure

1. Navigate to the **Contact→Search** menu item.
This displays the **Search Contacts** screen. You can also use the **Search→Contacts** menu item to display this screen.
2. Choose **Company** or **Person** from the **Type** drop-down list.
 - If you select **Company**, PolicyCenter displays a **Company Name** field.
 - If you select **Person**, PolicyCenter displays a **First name** and **Last name** field.

Contact search results

If you are integrated with an external contact management system, **Contact→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. You can modify the code that specifies the tag returned by ContactManager. See the description of `ABCContactAPISearchCriteriaEnhancement` in the *Guidewire Contact Management Guide*.

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

- *Guidewire Contact Management Guide*

Selecting recently viewed contacts

Recently viewed contacts appear at the bottom of the **Contact** drop-down menu. Select one of these contacts to view the **Contact File Details** for that contact. If the contact is an account holder, PolicyCenter displays the **Account Holder Summary** screen.

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 42.

View contact file details

About this task

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**.

Procedure

1. 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.

2. Click **Edit Contact** to make changes.

Create new account from contact

Procedure

1. Select **Actions**→**New Account** to create an account with the current contact as the account holder.

PolicyCenter displays the **Create account** screen. This screen is the same screen that PolicyCenter displays when you select **Account**→**New Account** then select **Create New Account**. PolicyCenter populates most of the fields on the **Create Account** screen with values from the contact. You must select a producer.

2. 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

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**.

View policies associated with a contact

Procedure

1. While 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. This screen displays the accounts and policies for which you have sufficient producer code permissions.

2. 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.

3. Filter the list 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, then **PolicyPeriod** has a **PolicyContactRole** that points to the current **Contact** through the **ContactDenorm** field.

View policy transactions associated with contact

Procedure

1. While 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.

2. Click the **Transaction #** link for detailed information.

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.

3. 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 in the list.

4. 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.

View claims associated with a contact

Procedure

1. While 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`.

2. 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.

Contact File Claims screen

On the **Contact File Claims** screen, 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

- *Integration Guide*
- *Installation Guide*

Viewing billing information for a contact

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 columns that contain PolicyCenter information are:

- **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 information

Some columns come from billing account information returned by the billing system. These columns are:

- **Next Invoice Due** – When the next invoice is due and the amount.
- **Last Payment** – The amount of the last payment.
- **Billing Account** – The name on the account.

If the BillingCenter integration is enabled, this is a link to the billing account or subaccount in BillingCenter.

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`.
- **Status** – Displays **Delinquent** in red or **In Good Standing**.
- **Billed Outstanding** – The billed outstanding amount.

- **Current Due** – The current amount due.
- **Past Due** – The amount past due.
- **Unbilled** – The amount not billed.

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** or **Account Holder Summary** 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

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 "Create a new contact" on page 358.

Add a contact to an account

Procedure

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 363
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 **Contacts** 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.

Add a contact from the address book

About this task

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 contact types are the **ABPerson** and **ABCCompany** subtypes of **ABContact**.

Procedure

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**.
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.

Edit a contact

About this task

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 355.

To modify contact information on an account:

Procedure

1. On the **Account File Contacts** screen, select  to edit the chosen contact.

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**.

Remove a contact from an account

About this task

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.

To remove a contact from an account:

Procedure

1. On the **Account File Contacts** screen, select the check box of the contact that you want to remove.
PolicyCenter prevents you from removing the account holder and contacts that are associated with any policy or policy transaction.
2. Click **Remove Contact**.

Change the active status on a contact from an account

About this task

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:

Procedure

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**.
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

Add or remove a contact role

Procedure

1. On the **Account File Contacts** screen, click  for the contact that you want to change.
2. On the **Roles** tab, 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 new or existing contact has the same address as another contact in the same account, you can link the addresses to avoid retying the address. In the base configuration, the linked-to contact must be a primary named insured, named insured, or account holder. If you edit the address, you can update all linked addresses or update only the current address and unlink.

See also

- “Linking an address between multiple contacts” on page 357
- “Linked addresses object model” on page 372
- “Configuring linked addresses for contacts” on page 391

Link to address on another contact

About this task

You can

Procedure

1. In PolicyCenter, navigate to a contact that is an account file contact or create a new account file contact.
2. Edit the contact.
3. 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.
4. Select an address.
 - The address is copied into the address fields.
 - The address section displays **This address is linked to other addresses**.
5. Click **Update** to save your changes.

You can also link to an address on the **Addresses** tab of the contact.

Edit linked address

About this task

A linked address displays **This address is linked to other addresses** above the **Address** label.

Procedure

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.

Changing revised contact information in 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.

Modify revised information in the account contact

Procedure

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**.

Next steps

Next, create a future-dated change. See “Create a future-dated change to revised contact information” on page 367.

Create a future-dated change to revised contact information

Before you begin

You must first modify revised information in an account contact. See “Modify revised information in the account contact” on page 367.

About this task

Ray Newton has a bound personal auto policy. Recently, Ray called and had his first name on the account changed to Raymond. Because the policy was in force when he called, Ray's first name was not updated on the policy.

In these steps, you will create a future-dated policy change to revised contact information. You will change the contact's name and marital status.

Procedure

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.

The name of the **Primary Named Insured** has been changed to **Raymond Newton** because the contact information was copied from the account at the start of the policy transaction.

6. Advance to the **Drivers** screen and select the driver John Smith.

In the next step, you will make the future-dated change to revised contact information.

7. 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**.

Next steps

"View a future-dated change to revised contact information" on page 368

View a future-dated change to revised contact information

Before you begin

"Create a future-dated change to revised contact information" on page 367

About this task

These steps assume that you have created a future-dated change to revised contact information. By viewing the policy on a date after the change, you can see the change to the revised contact information.

Procedure

1. Navigate to the policy with the future-dated change.
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 revised contact information is updated. If you changed the Ray Newton policy as described in the previous steps, the driver John Smith is now 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 revised contact information is not updated. If you changed the Ray Newton policy, the driver's name is still John Smith and he is single.

Next steps

"Update account with future-dated change to revised policy contact information" on page 368

Update account with future-dated change to revised policy contact information

Before you begin

This procedure assumes that you have verified a future-dated change to revised contact information. See "View a future-dated change to revised contact information" on page 368.

About this task

Previously, you updated revised contact information in a policy change. In these steps, you move the system clock forward past the date of the change. Then you run the Apply Pending Account Updates batch process to apply the change to the account contact.

To update the account contact with future-dated changes to revised contact information, you must move the system clock. To move the system clock, you must enable the **Internal Debug** tools.

Procedure

1. Navigate to the Raymond Newton account. This account is the account of the policy with changes to revised contact information.
2. In the left sidebar, click **Contacts**.
3. 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.

Enable internal debug tools

4. In Studio, edit config.xml.
5. Check the value of the EnableInternalDebugTools parameter. Set it to true.
6. If you updated the EnableInternalDebugTools parameter, restart PolicyCenter.

Move the system clock forward

7. In PolicyCenter, log in as su.

The su has the necessary privileges for using the run command and for running the batch process.

8. 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.

Apply pending account updates

9. Press ALT+SHIFT+T to display **Server Tools**.
10. On the **Batch Process Info** screen, click **Run** in the **Apply Pending Account Data Updates** batch process.
In the base configuration, this batch process runs nightly.
11. Select **Actions→Return to PolicyCenter**.

View changes to revised contact information in the account

12. Navigate to the Raymond Newton account.

13. 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.

14. Click the contact that you changed.

Notice that the future-dated change to name and marital status are updated.

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.

Create back-dated change to revised contact information

Procedure

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 "Create back-dated change to revised contact information" on page 369. 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**.

Next steps

Next, you can view the back-dated change. See “View back-dated change to revised contact information” on page 370.

View back-dated change to revised contact information

Before you begin

This procedure requires that you have created a back-dated change. See “Create back-dated change to revised contact information” on page 369.

Procedure

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 **0/03/2011**.
5. In the left sidebar, click **Drivers**. The driver’s date of birth is updated.

Next steps

Next, you can create a back-dated change to a revised field. See “Create back-dated change to revised marital status” on page 370.

Create back-dated change to revised marital status

Before you begin

This procedure requires that you verified a first back-dated change to revised contact information. See “View back-dated change to revised contact information” on page 370.

Procedure

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.

Update the account contact with back-dated change to revised policy contact information

Before you begin

These instructions assume that you completed:

- “Changing revised contact information in future-dated policy change” on page 367
- “Changing revised contact information in a back-dated policy change” on page 369

Procedure

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.

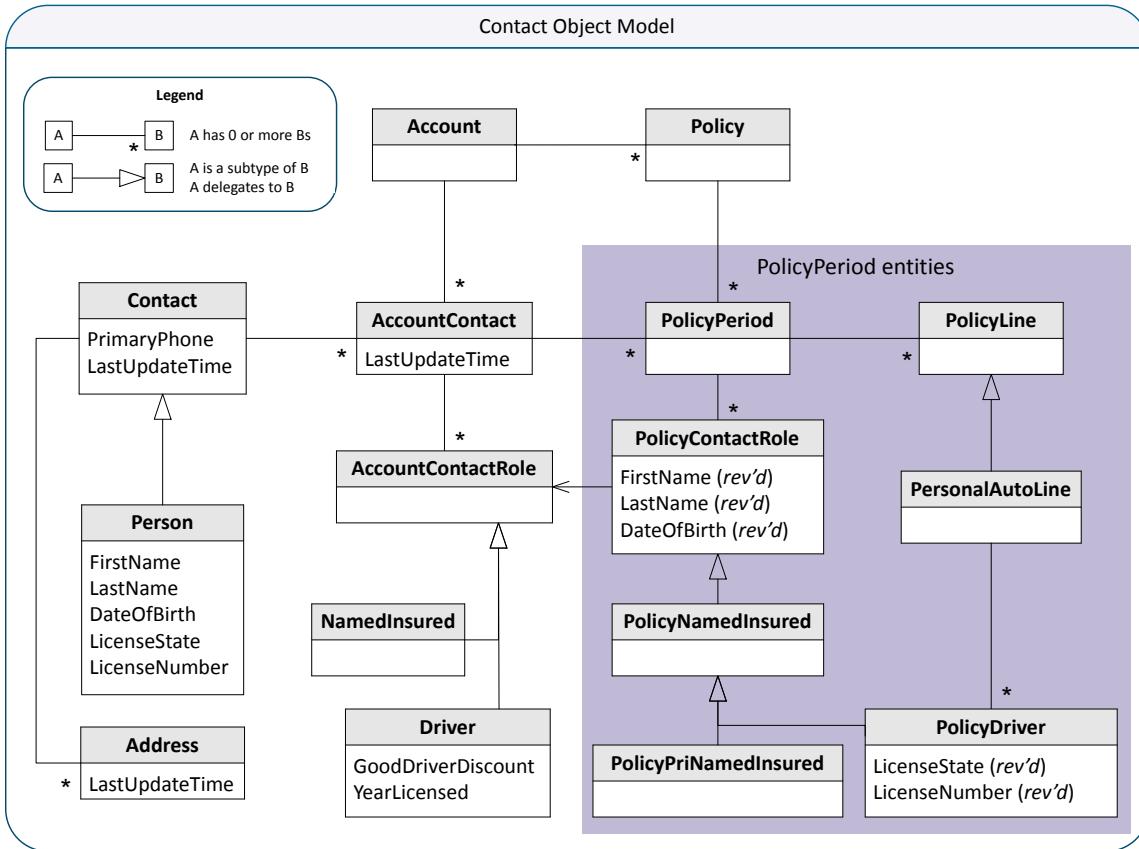
At the bottom of the screen, PolicyCenter displays the **Activity Detail**. The **Description** says that changes have only been applied to the policy. If you want the changes applied to the account level contact, you must do that manually. On the right side of the screen, the **Text** of the **New Note** 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.
 - a. Click **Edit**, and **Complete** the activity.
 - b. 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. For information on the Contact data model, see the *Guidewire Contact Management Guide*.

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 revisioned fields. These revisioned fields are marked *rev'd*. These fields are revisioned as described in “Revisioning contact information in policies” on page 355.

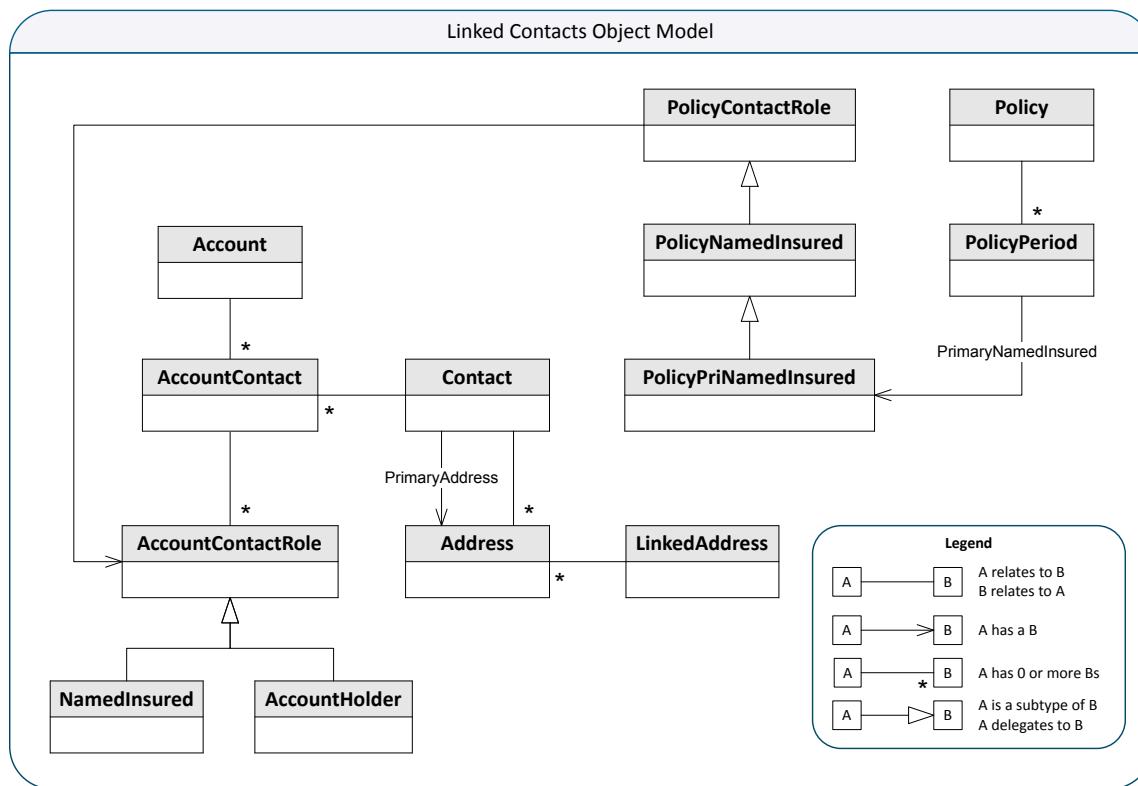
Revisioned fields implement the account syncable interface. For more information, see “Account synchronization classes for contacts” on page 374.

See also

- “Policy revisioning” on page 453
- *Guidewire 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 357
- “Working with linked addresses” on page 366
- “Configuring linked addresses for contacts” on page 391

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	

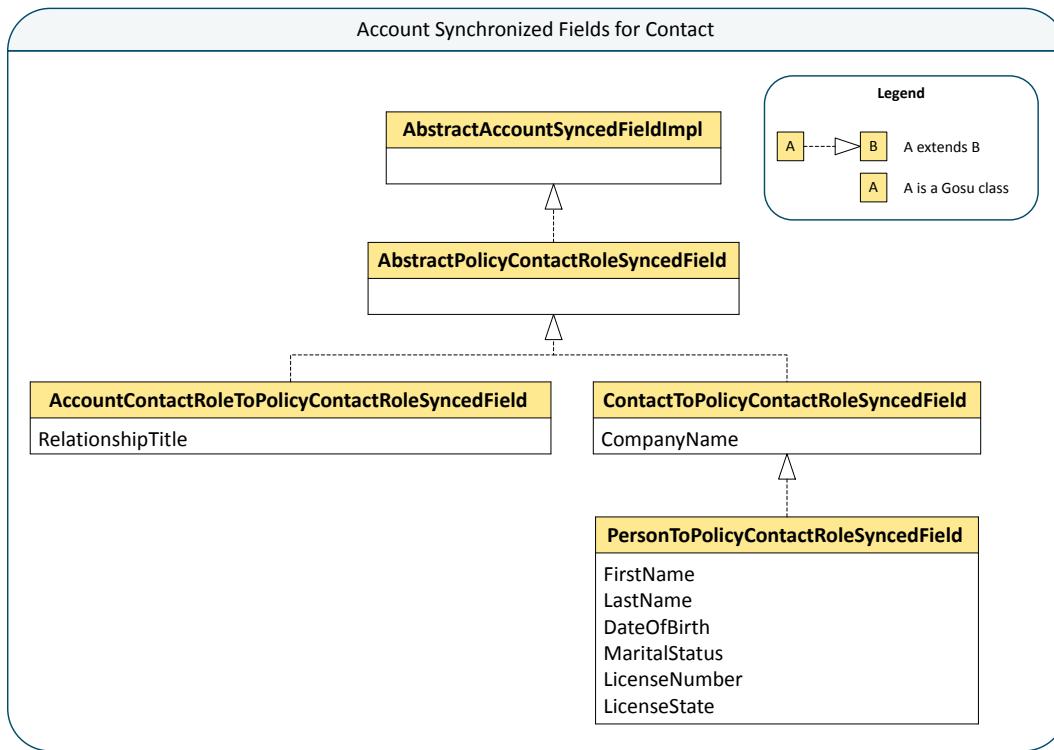
AccountContactRole subtype PolicyContactRole subtype	
InspectionContact	
AccountingContact	
NamedInsured	PolicyNamedInsured
Driver	PolicyDriver
BillingContact	PolicyBillingContact
AdditionalInsured	PolicyAddlInsured
AdditionalInterest	PolicyAddlInterest
SuppliedEmployee	PolicySuppliedEmployee
ReceivedEmployee	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.

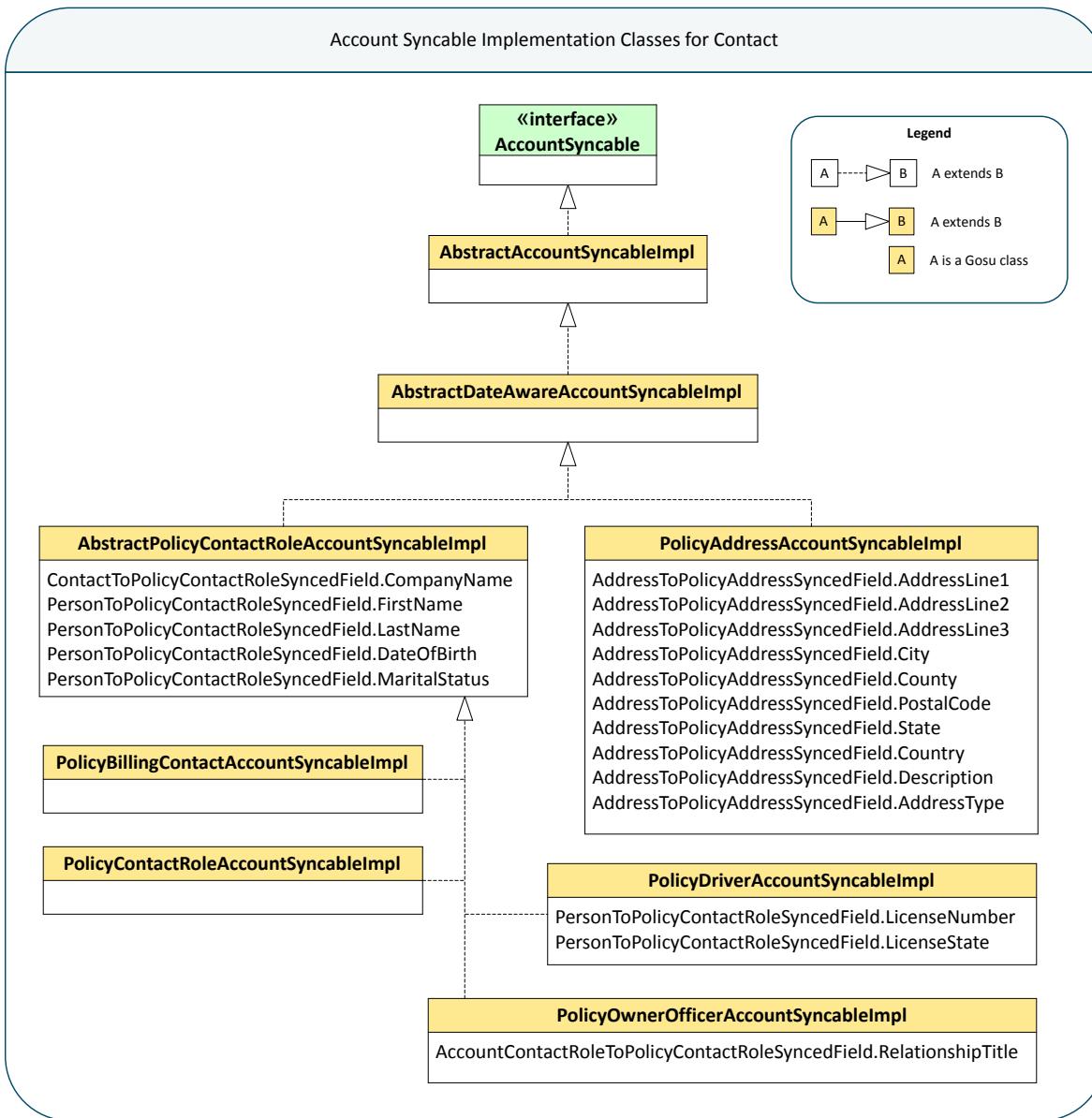
In some cases, a policy contact role is associated with an account level contact role, such as `PolicyDriver` and `Driver`. A policy contact role can be added to any of the policies within that account. (Many roles only apply to certain lines of business.) A contact role can contain fields that are shared 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 384
- 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 revisioned.
- 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.

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.

Configurable Gosu classes for revised fields

Policy period entities with revised fields have configurable Gosu classes 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 fields in the entity are updated by calling the getter and setter methods in the enhancement.

The following table lists some of these classes provided in the base configuration.

Files	Fields
PolicyContactRoleAccountSyncableImpl.gs PolicyContactRoleEnhancement.gsx	Synchronizes CompanyName, FirstName, and LastName for all policy contact roles. These files are in the gw.contact package.
PolicyDriverAccountSyncableImpl.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.
PolicyOwnerOfficerAccountSyncableImpl.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** plugin 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 base configuration, each **PolicyContactRole** references one **AccountContactRole**. However, you can configure several **PolicyContactRoles** to reference one **AccountContactRole**.

See also

- *Integration Guide*

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.

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 355
- *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 355
- *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 353
- “Working with the Contact tab” on page 357

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
<code>ContactAssociationFinder.gs</code>	Contains code for finding accounts, policies, claims, and billing accounts related to the contact file.
<code>ProductCodeFilterSet.gs</code>	Contains code for filtering products on contact file list views.
<code>JobFilters.gs</code>	Contains code for filtering status and policy transactions in contact file list views. See the <code>StatusFilterSet</code> and <code>JobTypeFilterSet</code> methods.
<code>PolicyPeriodBaseEnhancement.gsx</code>	Contains code that gets the <code>PeriodDisplayStyle</code> . The Contact File Policies screen uses this status to filter policy periods.
<code>ClaimSearchCriteria.gs</code>	Contains code for searching on the Contact File Claims screen.
<code>ContactEnhancement.gsx</code>	Contains code that gets the <code>PolicyPeriods</code> 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

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.
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.

Configuring 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.

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:

- “Creating entities that define the new subtypes” on page 380
- “Create an implementation of the contact configuration plugin” on page 382
- “Add display key and entity name” on page 382
- “Modify PCF files and Gosu classes” on page 384

See also

- *Configuration Guide*

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

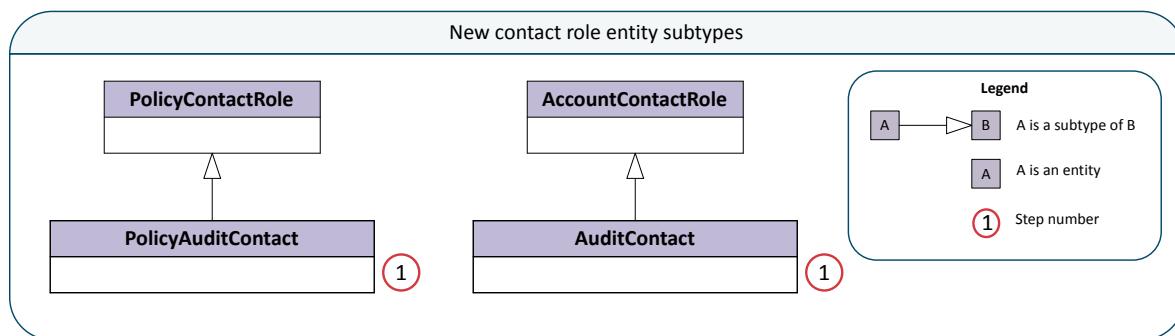
Creating entities that define the new subtypes

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 the *Configuration Guide*.

In this procedure, 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 extension files for audit contact

Procedure

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	subtype
Desc	Policy Audit Contact
Supertype	<code>PolicyContactRole</code>
<code>displayName</code>	<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`.

Next steps

“Create an implementation of the contact configuration plugin” on page 382

Create an implementation of the contact configuration plugin

Complete the step “Creating entities that define the new subtypes” on page 380 before you perform this step.

Create an implementation of the `IContactConfigPlugin` interface based on `ContactConfigPlugin.gw` in the `gw.plugin.contact.impl` package. Your implementation maps the new `AccountContactRole` subtype to the new `PolicyContactRole` subtype.

Implement Contact Configuration plugin

Procedure

1. In Studio, type `CTRL+N`, enter `ContactConfigPlugin`, and double-click to open `gw.plugin.contact.impl.ContactConfigPlugin`.
2. Select all the text except for the package name in `ContactConfigPlugin.gw` and copy it to the clipboard.
3. Navigate to the `gw.plugin.contact.impl` package in `configuration→gsrc`.
4. Right-click `impl` and select **New Gosu Class**.
5. In the **New Gosu Class** dialog, enter `ContactConfigPlugin_mine` in the **Name** field, then click **OK**.
6. Remove the class declaration, then paste the contents of the clipboard at the end of the file.
7. Change the class name from `ContactConfigPlugin` to `ContactConfigPlugin_mine`.
8. Add the following code to `protected property get DefaultConfigs`:

```
new ContactConfig(true, {TC_COMPANY, TC_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 `{TC_PERSON}`; or if only a company, use `{TC_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.

Next steps

“Update plugin registry” on page 382

Update plugin registry

About this task

“Implement Contact Configuration plugin” on page 382

Procedure

1. In Studio, navigate to `configuration→config→Plugins→registry` and open `IContactConfigPlugin.gwp`.
2. In `IContactConfigPlugin.gwp`, in the `Gosu Class` field enter the name of your plugin. If you are following this example, the plugin name is `ContactConfigPlugin_mine`.

Next steps

“Add display key and entity name” on page 382

Add display key and entity name

Complete the step “Create an implementation of the contact configuration plugin” on page 382 before you perform this step.

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.

Add a display key

Procedure

1. In Studio, navigate to **configuration**→**config**→**Localizations** and open `display_xx_XX.properties` for the default language.
2. Add the following line:

```
entity.AuditContact_Ext = Audit Contact
```

Next steps

[“Add an entity name” on page 383](#)

Add an entity name

Before you begin

[“Add a display key” on page 383](#)

About this task

Add an entity name for the `AuditContact_Ext` entity that is associated with the display key.

Procedure

1. In Studio, navigate to **configuration**→**config** and right-click **Entity Names** and select **New Entity Name**.
2. Enter `AuditContact_Ext` in the **Entity** field and click **OK**.
Studio opens `AuditContact_Ext.xx` for the default language.
3. In the editor, enter the following in the text field of the **Default** tab at the bottom of the screen.

```
entity.AuditContact_Ext
```

Next steps

[“Create contact that uses new contact role” on page 383](#)

Create contact that uses new contact role

Before you begin

[“Add an entity name” on page 383](#)

Procedure

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**.

Next steps

“Modify PCF files and Gosu classes” on page 384

Modify PCF files and Gosu classes

Before you begin

Complete the step “Add display key and entity name” on page 382 before you perform this step.

About this task

Change PCF files to configure how users 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 380.

You are done adding new contact roles.

Adding a revisioned field to a contact

This topic describes how to add revisioned 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 revisioned fields occurs. If a change occurs after this window, a batch process applies the future-dated change to the revisioned field. If the change occurs before this window, a user decides when to apply the back-dated change to the revisioned field. For more information, see “Revisioning contact information in policies” on page 355.

This topic provides general instructions and step-by-step instructions for adding a revisioned field to a contact. The step-by-step instructions guide you to add a `AuditLicenseNumber` revisioned field to the `AuditContact` entity. (For instructions on how to create the `AuditContact` and `PolicyAuditContact` entities, see “Configuring contact roles” on page 379.)

Follow these steps to add a revisioned field to `AuditContact`.

See also

- *Configuration Guide*

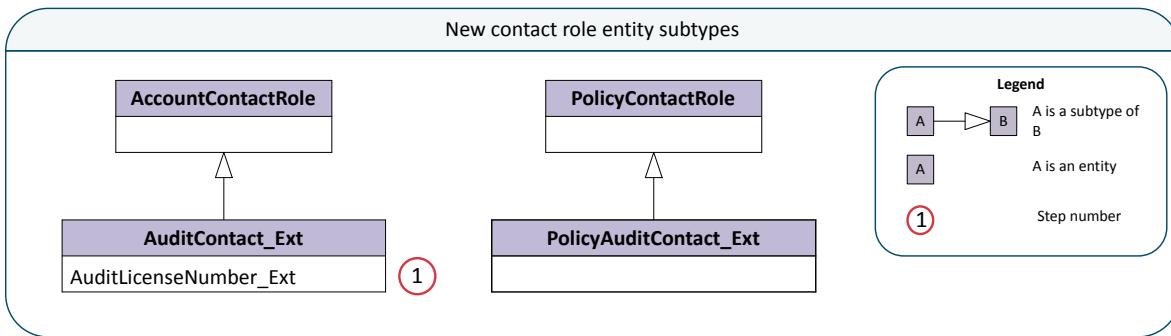
Defining the revisioned field on the account audit contact

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:

- If this is a new field, append `_Ext` to the field name avoid name conflicts with base PolicyCenter entities.
See the *Configuration Guide*.

In the following illustration, the circled number 1 shows the `AuditLicenseNumber_Ext` field that you will create on the `AuditContact_Ext` entity.



Add revised field to the account audit contact

About this task

Define the revised field for `AuditContact_Ext`.

Procedure

- In Studio, navigate to **configuration**→**config**→**Extensions**→**Entity** and double-click to open `AuditContact_Ext.eti`.
- Next to **+ New**, 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
name	<code>AuditLicenseNumber_Ext</code>
type	<code>mediumtext</code>
desc	The audit license number

Next steps

“Defining the revised field on the policy audit contact” on page 385

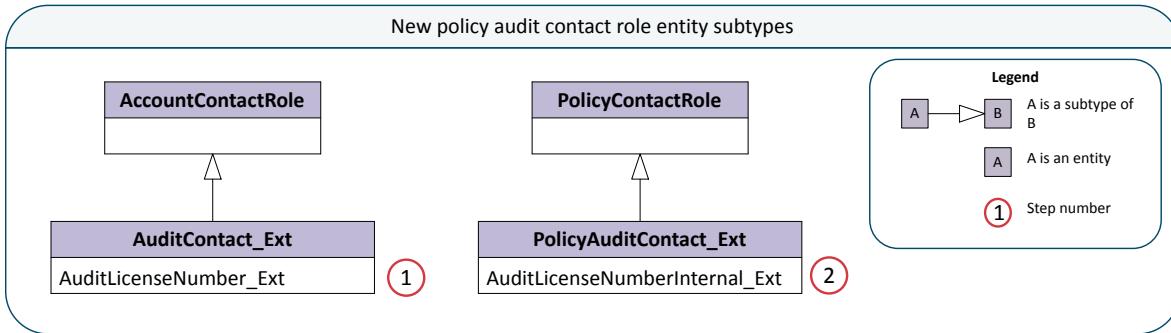
Defining the revised field on the policy audit contact

Complete the step “Defining the revised field on the account audit contact” on page 384 before you perform this step.

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 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.



Define revised field on policy audit contact

Procedure

- In Studio, navigate to `PolicyAuditContact_ext.eti` entity as you did in “Defining the revised field on the account audit contact” on page 384.
- Open `PolicyAuditContact_Ext.eti`.
- Next to , click the drop-down list and choose `column`.
- Enter the following values for the new column:

Name	Value
name	<code>AuditLicenseNumberInternal_Ext</code>
type	<code>mediumtext</code>
desc	The policy audit contact
getterScriptability	<code>doesNotExist</code>
setterScriptability	<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
iface	<code>gw.api.domain.account.AccountSyncable</code>
impl	<code>gw.contact.PolicyAuditContactAccountSyncableImpl</code>

The implementation does not exist yet. You add this implementation in later steps.

Next steps

“Defining field as syncable on the policy contact role” on page 386

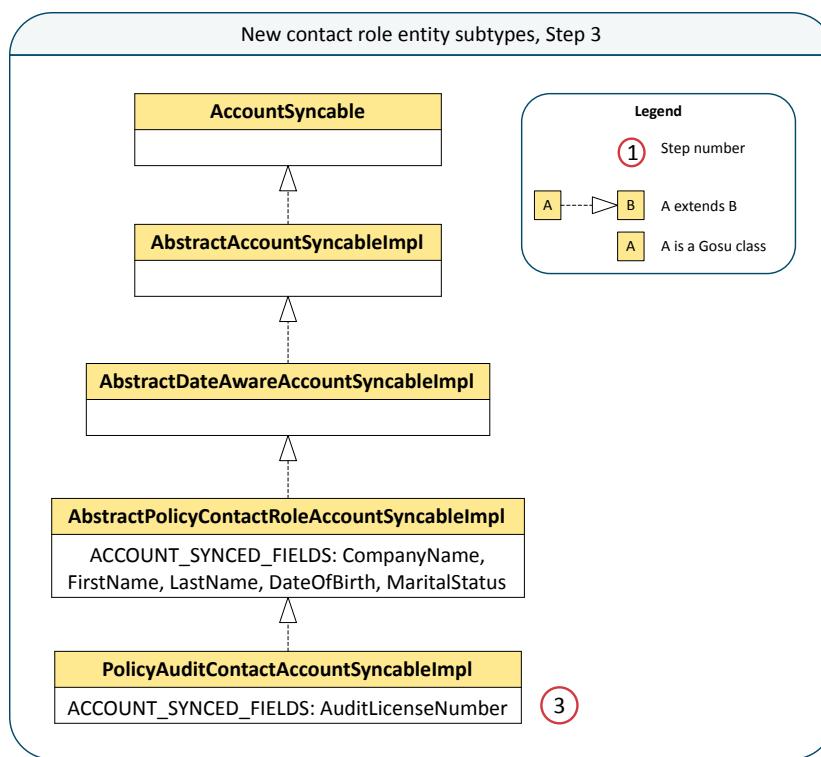
Defining field as syncable on the policy contact role

Complete the step “Defining the revised field on the policy audit contact” on page 385 before you perform this step.

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 355.



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`.

Define account syncable implementation class

Procedure

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**.

Next steps

“Define revised field as syncable on policy contact role” on page 387

Define revised field as syncable on policy contact role

Before you begin

“Define account syncable implementation class” on page 387

Procedure

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> {
    construct(accountSyncable: PolicyAuditContact_Ext) {
        super(accountSyncable)
    }
    static final var ACCOUNT_SYNCED_FIELDS = ImmutableSet.copyOf(
        AbstractPolicyContactRoleAccountSyncableImpl.AccountSyncedFieldsInternal.union(
            {
                AccountContactRoleToPolicyContactRoleSyncedField.AuditLicenseNumber})
    )

    protected static property get AccountSyncedFieldsInternal() :
        Set<AccountSyncedField<AccountSyncable, Object >> {
        // provided so subclasses can extend this list
        return ACCOUNT_SYNCED_FIELDS
    }

    override property get AccountSyncedFields() : Set<AccountSyncedField<AccountSyncable, Object >> {
        // must override to ensure that we call the correct static AccountSyncedFieldsInternal property
        return AccountSyncedFieldsInternal
    }
}

```

Note: You define AuditLicenseNumber in the next step.

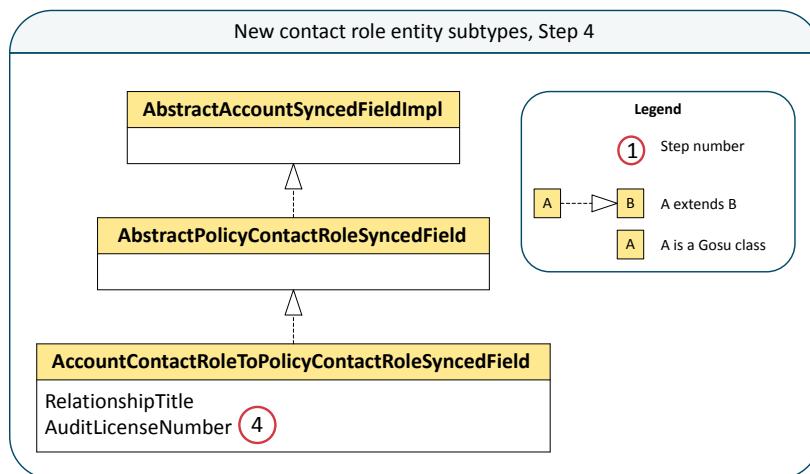
Next steps

“Define the field as syncable on the account contact role” on page 388

Define the field as syncable on the account contact role

Complete the step “Defining field as syncable on the policy contact role” on page 386 before you perform this step. Add the new field to the `gw.account.AccountContactRoleToPolicyContactRoleSyncedField` class. Define the new field as revisioned 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`.

Define revised field appended with `Internal_Ext`

Procedure

1. In Studio, open `gw.account.AccountContactRoleToPolicyContactRoleSyncedField.gs`.
2. Define a constructor for `accountEntityFieldName` with the following arguments:

```
class AccountContactRoleToPolicyContactRoleSyncedField<S extends PolicyContactRole, T>
    extends AbstractPolicyContactRoleSyncedField<String> {
    ...
    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.

3. Add the `AuditLicenseNumber` field to the class:

```
public static final var AuditLicenseNumber :
    AccountContactRoleToPolicyContactRoleSyncedField<PolicyAuditContact_Ext, String> =
    new AccountContactRoleToPolicyContactRoleSyncedField<PolicyAuditContact_Ext, String>
        ("AuditLicenseNumber", "AuditLicenseNumberInternal_Ext")
```

Next steps

“Extend entity and Gosu class for future-date policy changes” on page 389

Extend entity and Gosu class for future-date policy changes

Complete the step “Define the field as syncable on the account contact role” on page 388 before you perform this step.

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 355.

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.

Extend entity Gosu class

Procedure

1. Navigate to `configuration`→`config`→`Extensions`→`Entity`.
2. Right-click `Entity` and select `New`→`Entity Extension`.
3. Enter `PendingAccountContactRoleUpdate` and click `OK`.
4. In the extension file, add a column with the following values:

Name	Value
name	<code>AuditLicenseNumber_Ext</code>

Name	Value
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.

Next steps

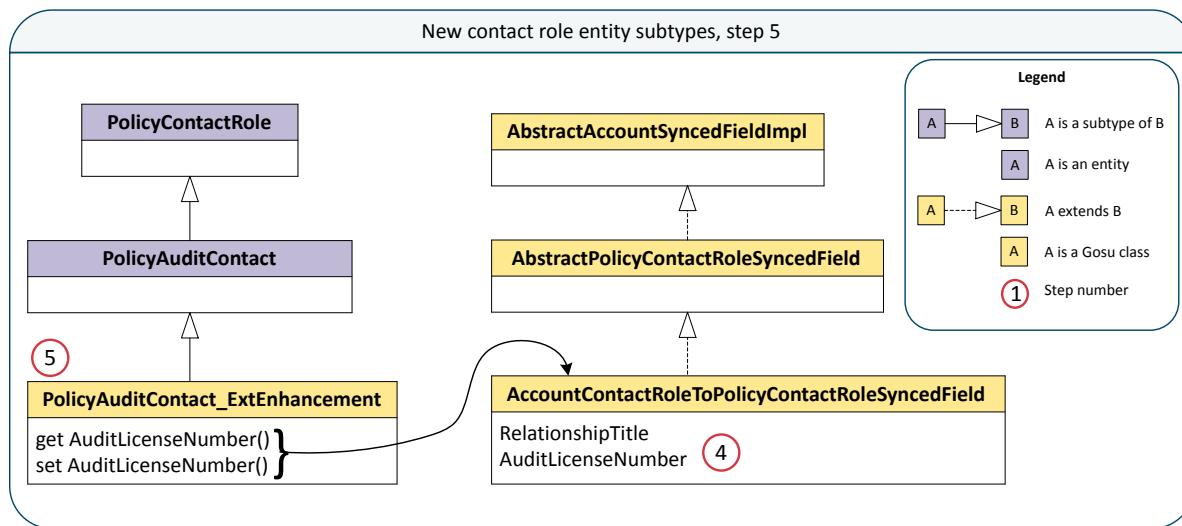
“Add get and set methods to the policy contact role” on page 390

Add get and set methods to the policy contact role

Complete the step “Extend entity and Gosu class for future-date policy changes” on page 389 before you perform this step.

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 methods access the `AccountContactRoleToPolicyContactRoleSyncedField` class. In the following illustration, the circled number 5 shows the get and set methods on the `PolicyAuditContact_ExtEnhancement` class.



Create new enhancement for contact role

Procedure

- In Studio, navigate to the `gw.contact` package.
- Right-click contact and select **New→Gosu Enhancement**.
- 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`.

Next steps

“Add get and set methods to the policy location” on page 391

Add get and set methods to the policy location

Before you begin

“Create new enhancement for contact role” on page 390.

Procedure

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 revisioned AuditLicenseNumber.  
 */  
property get AuditLicenseNumber() : String {  
    return AccountContactRoleToPolicyContactRoleSyncedField.AuditLicenseNumber.getValue(this)  
}  
  
/**  
 * Shared and revisioned AuditLicenseNumber.  
 */  
property set AuditLicenseNumber(arg : String) {  
    AccountContactRoleToPolicyContactRoleSyncedField.AuditLicenseNumber.setValue(this, arg)  
}
```

Next steps

“Add revised field to PolicyCenter user interface” on page 391

Add revised field to PolicyCenter user interface

Complete the step “Add get and set methods to the policy contact role” on page 390 before you perform this step.

To see this field in PolicyCenter, you must add it to various PCF pages.

You are done adding a revised field to a contact.

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 357
- “Working with linked addresses” on page 366
- “Linked addresses object model” on page 372
- *Integration Guide*

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 511.

See also

- The *Integration Guide* for information on the rating plugins and how to integrate your own rating engine with PolicyCenter.
- “Rating overrides” on page 411
- “Rating Management” on page 511
- *Configuration Guide*

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 cost 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 transaction 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. Onset transactions point to costs in the same period. Offset 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.

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 find this interface in Guidewire Studio by going to **configuration**→**gsrc** and navigating to the `gw.lob.LOB.financials` package.
- Can calculate the prorated amount from term amount and effective date.
- Provides additional Gosu functionality defined by the *LOBCostMethods* Gosu interface. This interface is 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.

Property	Description
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.

In the default configuration, a cost can be either prorated or flat.

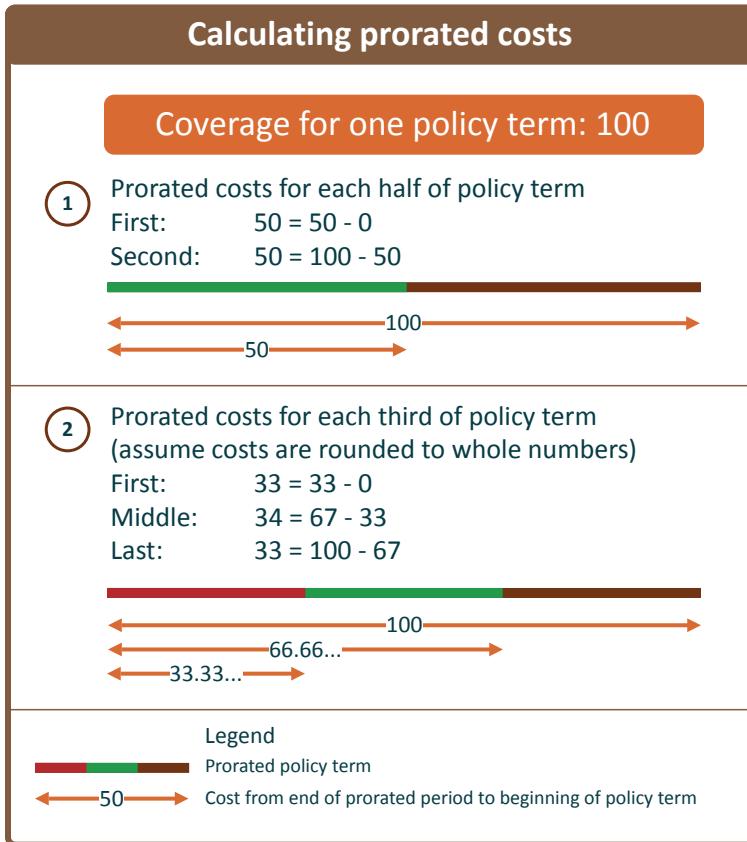
Prorated costs

The value of a prorated cost is calculated based on the number of days it exists on the policy. Generally, the cost is prorated by dividing the number of days in the 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 is sliced into two policy periods because a midterm policy change effective halfway through the policy term removes the coverage. The number of days in both policy periods is equal to half the term. The prorated cost for the coverage on the first half of the policy term is half of \$100, or \$50 for the policy period that it is in effect.

Sometimes prorating results in a cost that requires rounding. If the \$100 coverage cost on the policy term is sliced into three equal policy periods, dividing by three results in a cost of \$33.333... for each prorated policy period. Adding the three costs results in a value of \$99.99, not \$100. To ensure that the sum of the prorated costs always adds up to 100% of the unprorated cost, the prorated cost is computed using the following formula:

```
prorated cost = round(cost from end of prorated period to beginning of policy term)
               - round(cost from beginning of prorated period to beginning of policy term)
```

The following illustration shows how PolicyCenter calculates prorated costs. The costs are in dollars. For simplicity, assume that costs are always whole dollar amounts.



Example 1 shows a coverage that is effective for half of the policy term. If the coverage is in effect for the first half, the cost from end of prorated period to start of policy term is \$50. The cost from the beginning of prorated period to beginning of policy term is \$0. Rounding does not affect either value. The prorated cost is \$50 (\$50 minus \$0).

Example 2 shows a coverage that is effective for a third of the policy term. For each prorated policy period, the cost is \$33.33.... If the coverage is in effect for the middle third, the cost from end of prorated period to start of policy term is \$66.66.... When rounded to a whole number, the cost is \$67. The cost from the beginning of prorated period to beginning of policy term is \$33.33.... When rounded, the cost is \$33. The prorated cost of the coverage on the middle third of the policy is \$34 (\$67 minus \$33).

Flat costs

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 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).

Flat cost with price change

In the base configuration, flat cost rating is designed with the assumption that the term amount remains constant for a policy term, as described above. There are no circumstances that affect the price.

If in your implementation, the price of the flat cost is not constant, but can fluctuate between issuance and a later policy change, you will observe the following behavior.

A policy change that affects the price of a flat cost generates three transactions. The first transaction reverses the flat cost. The second applies the original flat cost for the old date range. The third transaction applies the new flat cost for the new date range. This results in two flat charges: the full amount for the old date range and the full amount for the new date range.

If you need other behavior, you can configure a custom proration method, **ProrationMethod**.

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 “In rate routine, specify coverage as flat-rated” on page 578.

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 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.

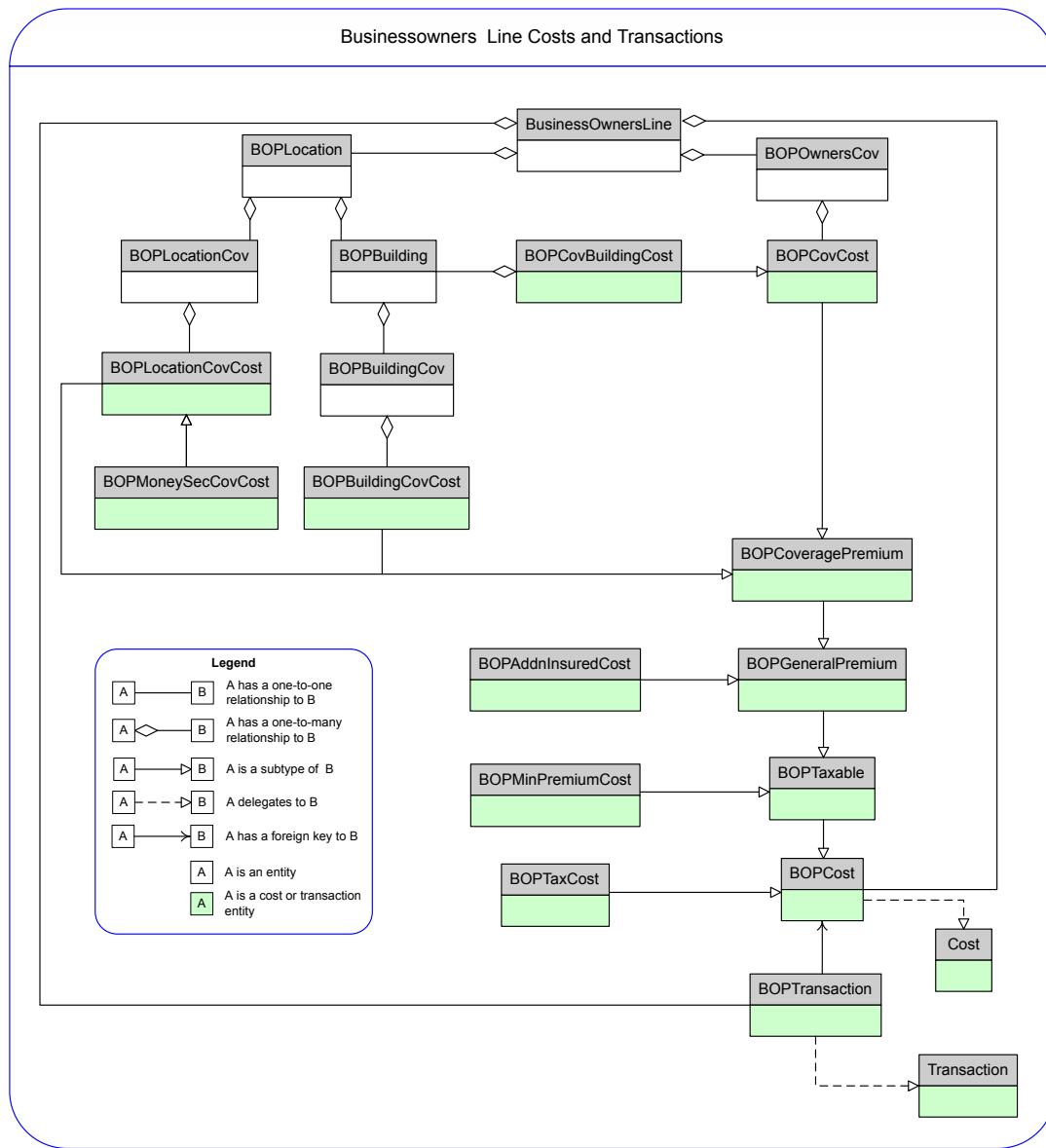
Cost and transaction model for businessowners line

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 **BOPTransaction** entity has a foreign key to **BOPCost**.

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**
- **BOPAddnInsuredCost**
- **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

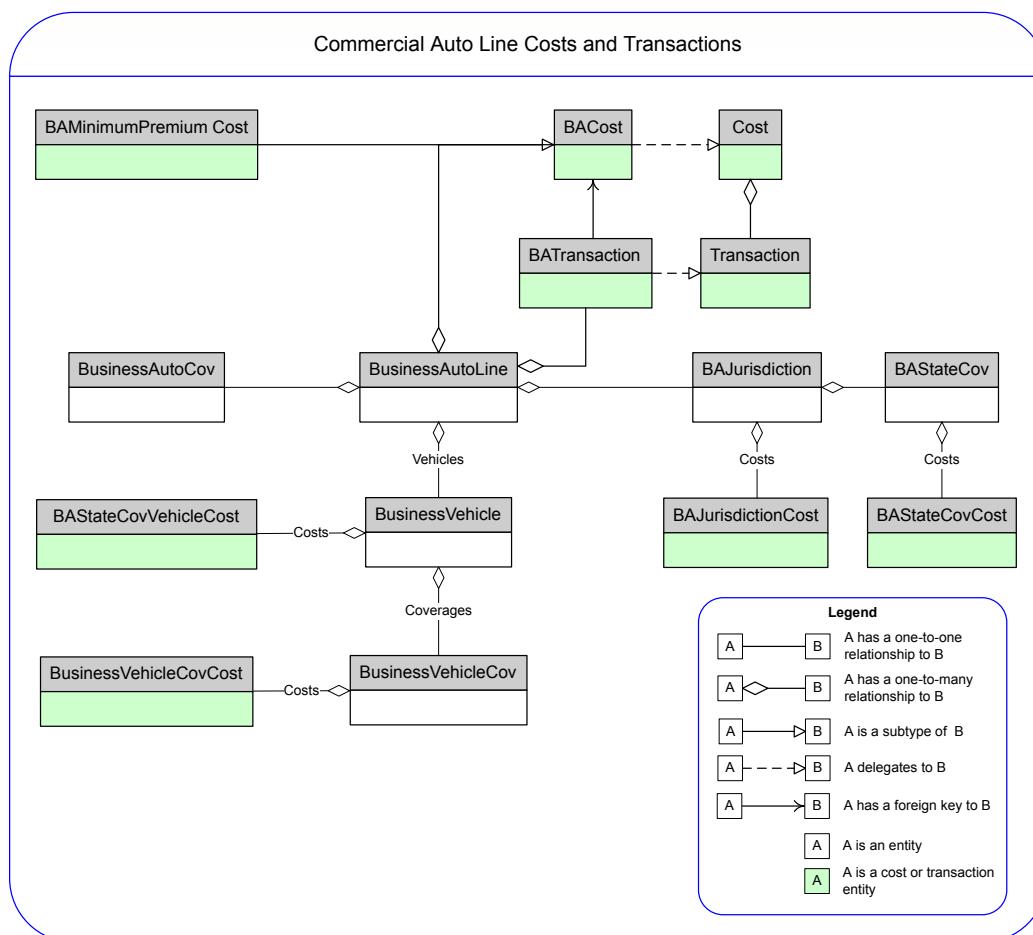
BOPCoveragePremium and BOPGeneralPremium costs. The cost supertype, **BOPCost**, includes costs of all abstract subtypes.

Cost and transaction model for commercial auto line

The following diagram illustrates how the various types of costs and transactions interact in the commercial auto line. The main entity for the commercial auto line is **BusinessAutoLine**, a subtype of **PolicyLine**. As you can see in the diagram, **BusinessAutoLine** connects directly to financial array **BACost**. The **BATransaction** entity has a foreign key to **BACost**.

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.

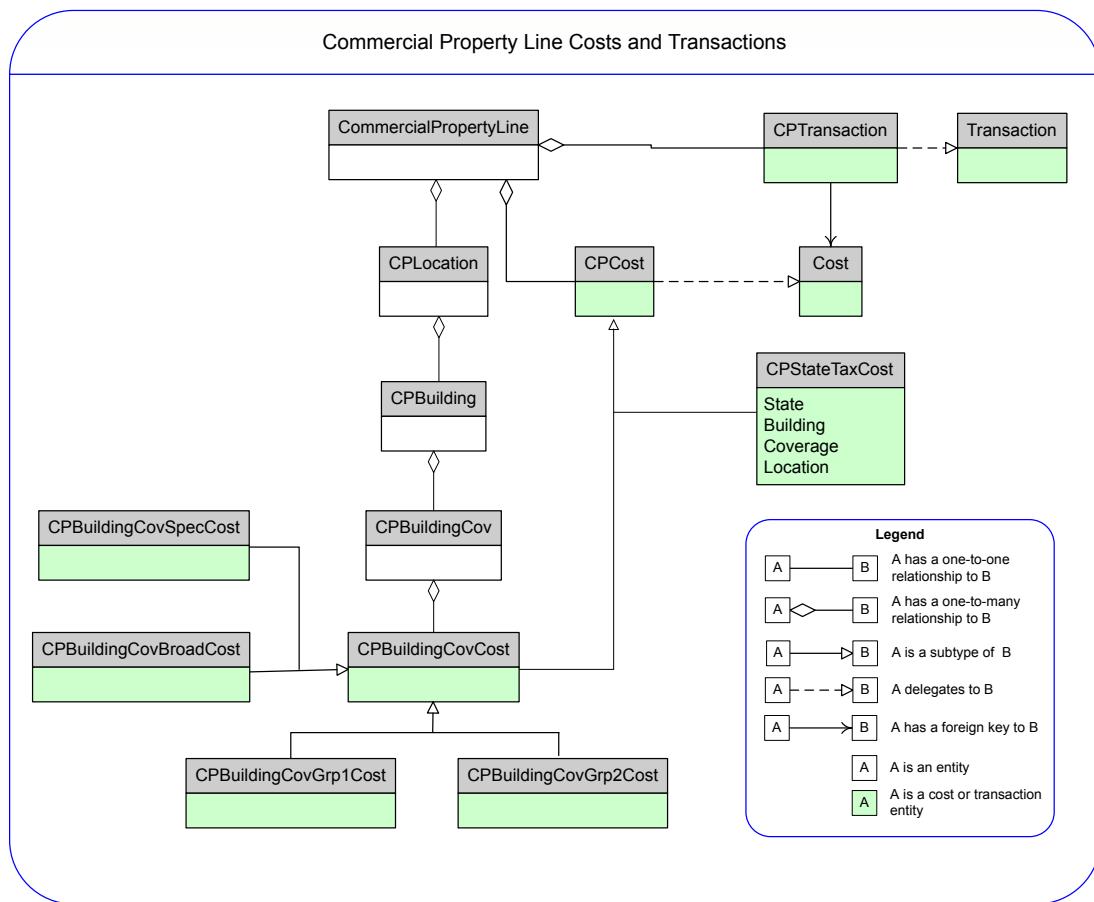


Cost and transaction model for commercial property line

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 **CPTTransaction** entity has a foreign key to **CPCost**.

The `CommercialPropertyLine` entity has a derived array to `CPTransaction`. The `CommercialPropertyLine` gets its derived array of transactions by asking the `PolicyPeriod` for all of its `CPTransaction` entities. The `PolicyPeriod` entity has an array key to `CPTransaction`.

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.



CPCost

The `CPCost` entity defines a cost associated with the commercial property line. The `CommercialPropertyLine` entity has an array of `CPCost` 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

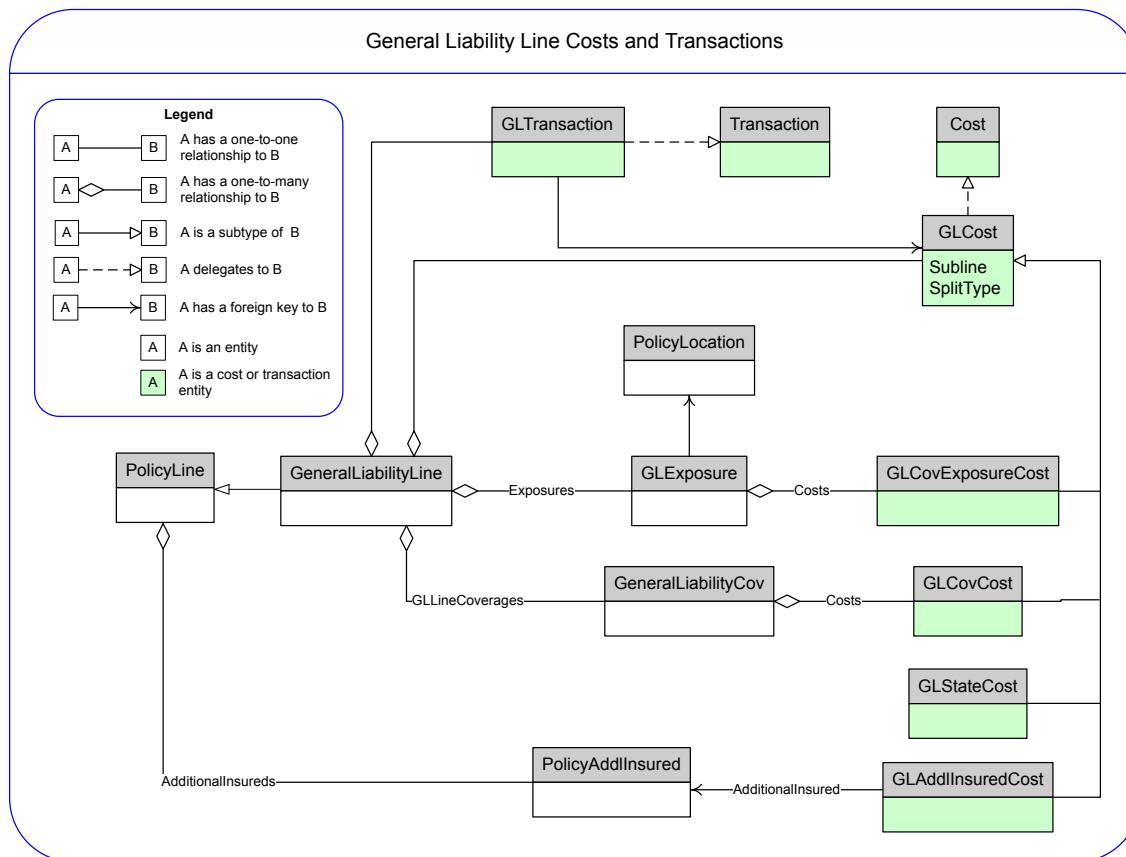
Captures the tax cost on the line per jurisdiction.

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 `GLTransaction` entity has a foreign key to `GLCost`.

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

- “Covrances screen for general liability” on page 213 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.

- Combined single limit for the premises subline
- Combined single limit for the product subline

If the user chooses to split bodily injury and physical damage limits, the rating engine creates four costs for each exposure.

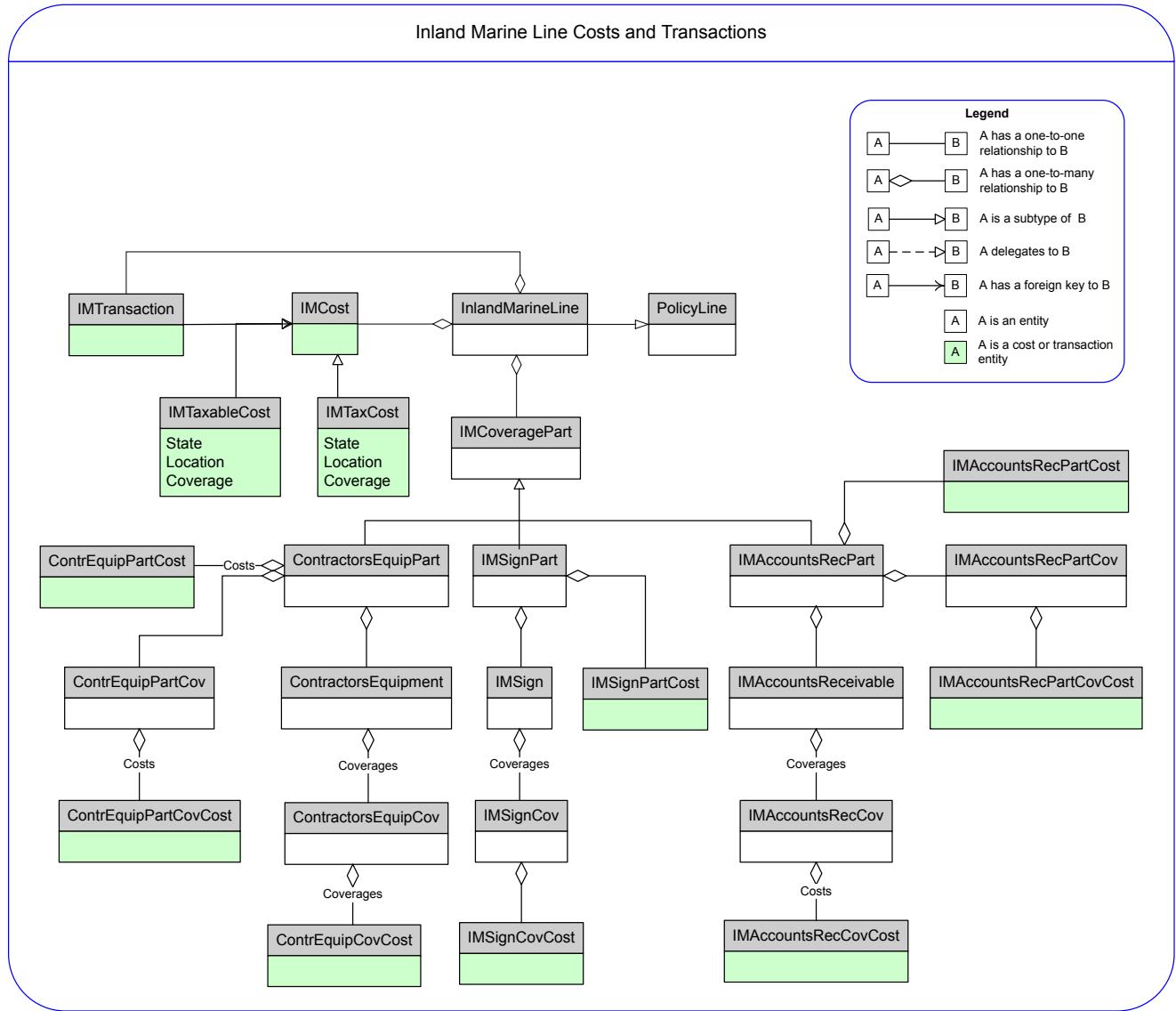
- Bodily injury for the premises subline
- Physical damage for the premises subline
- Bodily injury for the product subline
- Physical damage for the product subline

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 **IMTransaction** entity has a foreign key to **IMCost**.

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 transactions 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

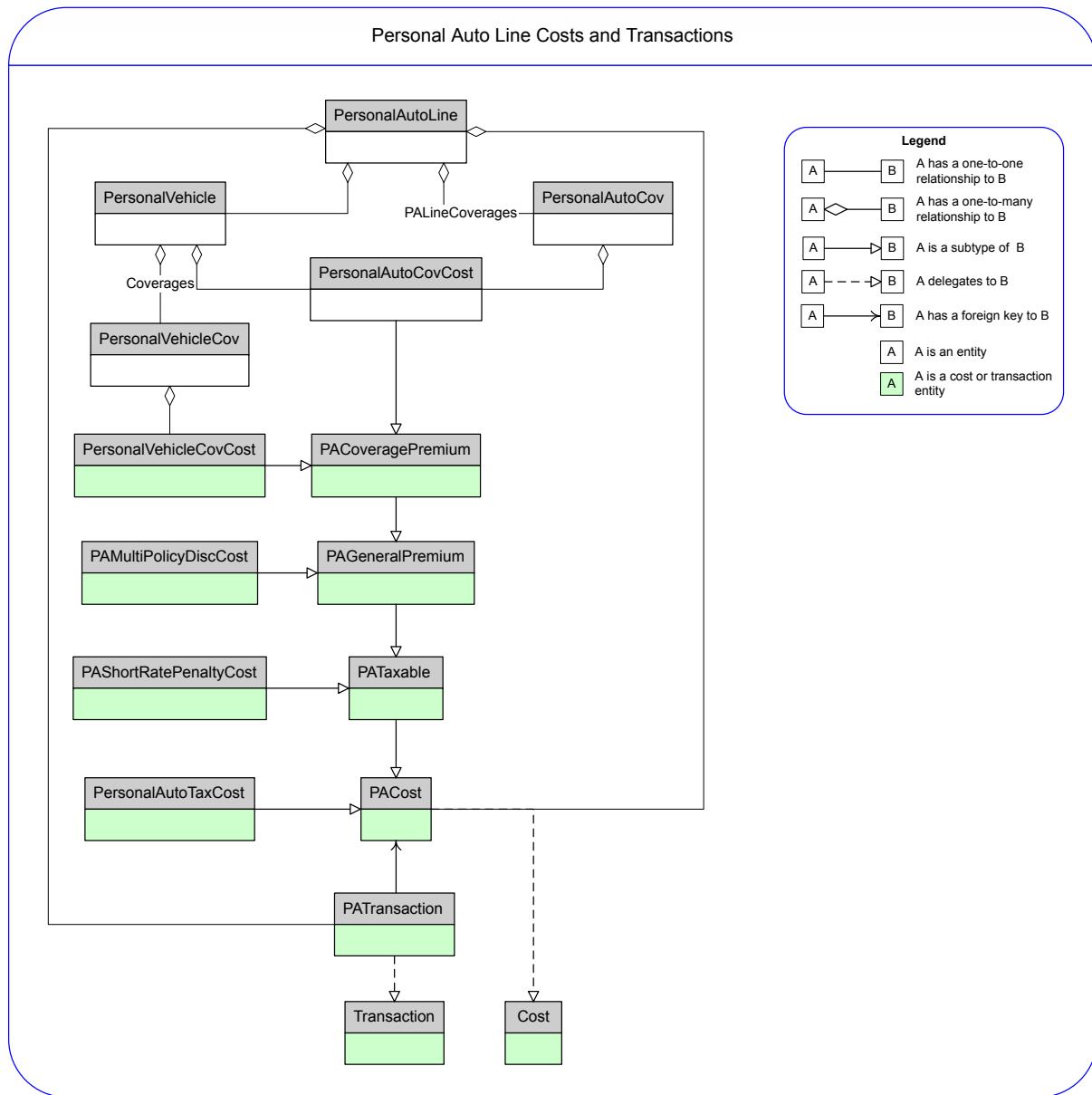
The following diagram illustrates how the various types of costs and transactions interact in the personal auto line. The PersonalAutoLine is a subtype of PolicyLine. As you can see in the diagram, PersonalAutoLine connects directly to financial arrayPACost. The PATransaction entity has a foreign key to PACost.

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

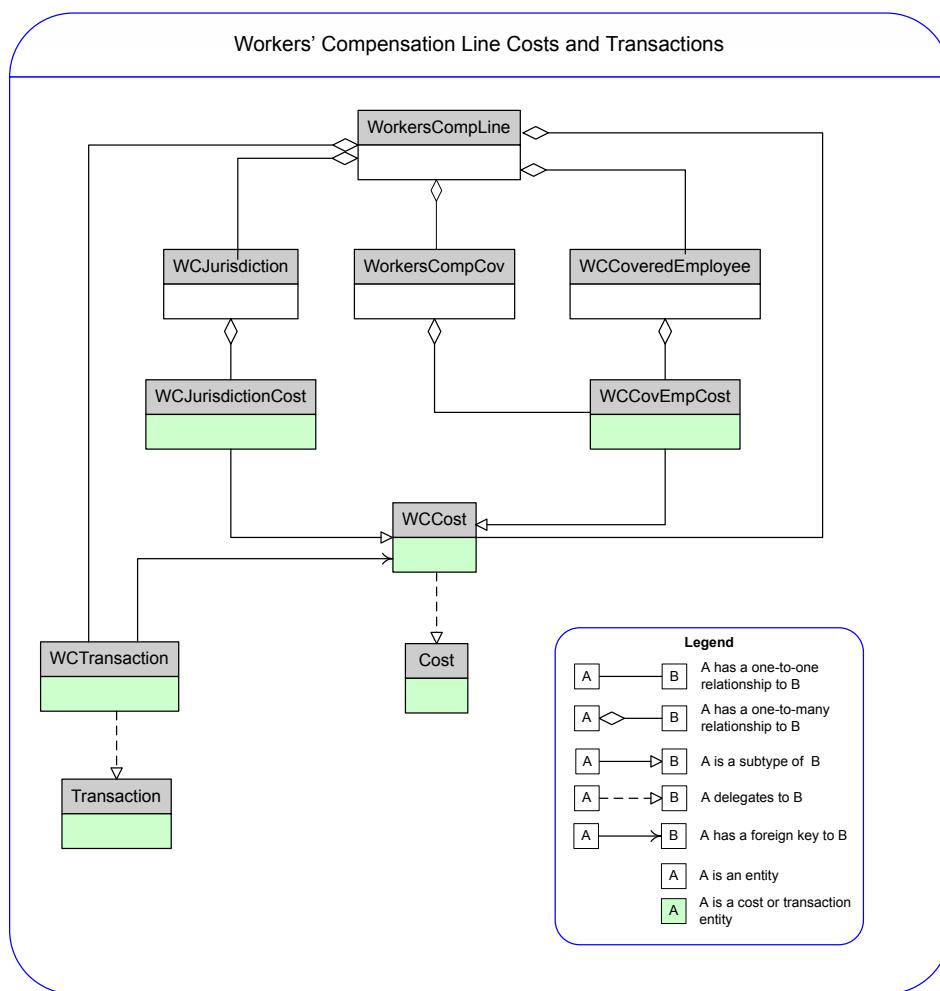
The abstract cost subtypes extend one another in the hierarchy. This hierarchy makes it easy to get all costs of at different levels of the hierarchy. For example, if you get all PAXTaxable costs, you also get all PACoveragePremium and PAGeneralPremium costs. The cost supertype, PACost, includes costs 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 WCTransaction entity has a foreign key to WCCost.

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 765.

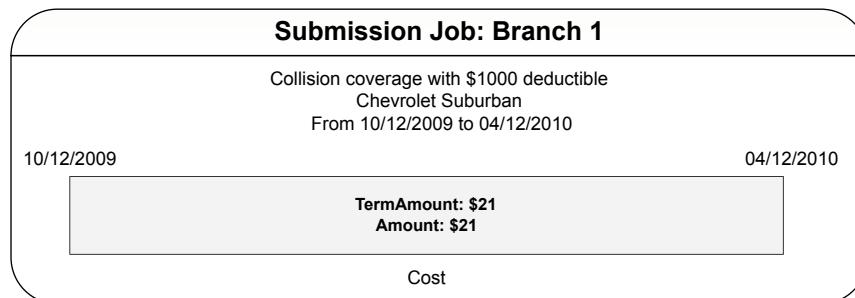
This topic provides an example showing costs and transactions in a personal auto submission policy transaction followed by a policy change transaction. The policy period is six months. In the submission, the customer selects collision coverage with a \$1000 deductible. Later, the customer calls and asks 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 a submission policy transaction effective August 13 to February 13, you create a six-month policy choosing Collision Coverage with a \$1000 Collision Deductible.

The **Policy Premium** tab on the **Quote** screen, the cost for collision coverage is \$21 and taxes are \$49.

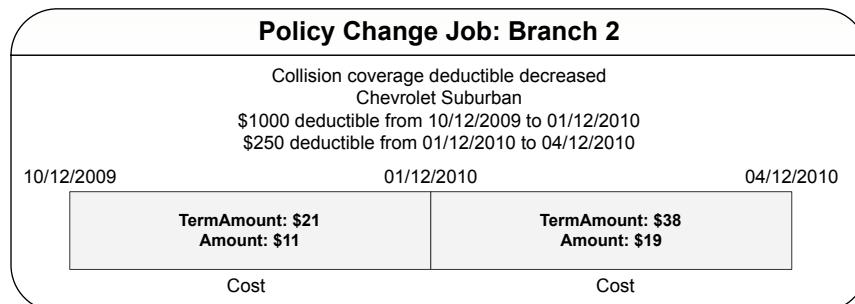
PolicyCenter creates a policy period branch with the cost for collision coverage as shown in the following diagram.



On the **Summary** screen, **Total Cost** shows the sum of all transactions for the submission as \$722.

At a later time, the insured calls to request a decrease in the Collision Coverage deductible from \$1000 to \$250 beginning November 13, three months from the policy effective date. The agent starts a policy change. The **Policy Premium** tab on the **Quote** screen displays the Premium for Collision Coverage. For the whole term, collision coverage with \$1000 deductible is \$21, and with \$250 deductible is \$38. The prorated cost is \$11 for the first three months, and \$19 for second three months.

PolicyCenter creates a new branch for the policy change. The new branch has two costs for collision coverage. Each cost has a prorated amount.



The database contains two costs as follows:

FixedID	ID	EffDate	ExpDate	Term	Amount	Car	Coverage
amount							
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. For Collision Coverage with \$1000 deductible, there is a (\$10.00) offset transaction in the **Premium** column. For Collision Coverage with \$250 deductible, there is a \$19.00 onset transaction.

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.

Transaction Calculation: Transactions			
	10/12/2009	01/12/2010	04/12/2010
Previous Branch		Amount: \$21	
Current Branch		Amount: (\$10)	Amount: \$19

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.

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.

Field	Description
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 39

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 insurer 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.

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	This permission allows you to: <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. The code for this permission is <code>editratingoverrides</code>.</p>
View rate and premium overrides	This permission allows you to: <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>

Permission	Description
	The code for this permission is <code>viewratingoverrides</code> .

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 issues” on page 671
- *Configuration Guide*

Processing rating 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 transaction, 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 included topics describe screens that display rating override information.

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 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.

Override a rating

About this task

This task assumes that you are in a policy transaction. To follow along, you can use policies in the small sample data set. Find a workers' compensation policy for Wright Construction, start a policy change transaction, and quote it.

Procedure

- On the **Quote** page, click the **Override Rating** button to display the **Rating Overrides** screen.
- Enter a value in the **Adjusted Rate** field. In this example, enter **4.0** in the **Adjusted Rate** for **Nurseries—propagation** in the first **Standard Premium** table. Optionally, enter a reason, such as **Low risk**, in the **Reason** field.
- Click the **Rerate** button.

This action sends the new values off to the rating engine.

The **Quote** screen appears and displays a warning message that there are overrides. This message 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 also input into the **Other Premium and Surcharges** table.

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.

5. In the **Override** columns, 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** screen appears again. Because you entered a flat amount, the **Basis** and **Rate** fields for the nurseries class code have no value.

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.

Enable Override Rating button

About this task

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**.

Procedure

1. In Studio, open **gw.lob.*Lob*.LOBPolicyLineMethods.gs**, where *Lob* is the abbreviation for the line of business.
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  
}
```

Next steps

You can now “Create panel set for rating overrides” on page 414.

Create panel set for rating overrides

Before you begin

“Enable Override Rating button” on page 414

About this task

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.

Procedure

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.

Next steps

“Update the rating engine to handle overrides” on page 415

Update the rating engine to handle overrides

Before you begin, you must “Create panel set for rating overrides” on page 414.

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).

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. 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. Orphaned policy periods are 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 associated objects. Quote purging is not an end user feature. Quote purging is disabled in the base configuration.

See also

- “Side-by-side quoting” on page 153
- “Multi-version quoting” on page 161
- “Preempted jobs” on page 474
- *Configuration Guide*

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 `DoNotDestroy`.

Quote purging does not purge policy transactions with archived policy periods. Quote purging does not prune archived policy periods.

Guidewire does not support configuring quote purging to remove archived policy periods.

Quote purging 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.

Quote purging: 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.

In addition to removing policy periods, purging also removes the policy transaction. For example, if purging removes all policy transactions associated with a policy, it also removes the `Policy` object when a `Policy` only exists because of an unbound submission.

Pruning only removes policy periods associated with alternate versions; pruning does not remove the policy transaction.

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

- *Configuration Guide*
- *System Administration Guide*

Quote cloning for business intelligence

Some insurance companies have a need to capture information from quotes that users and other processes generate through the course of the day. Quote cloning provides one approach for retaining some of this information long enough to capture it for business intelligence (BI) needs.

See also

- *Configuration Guide*

Quote cloning overview

Some insurance companies have a need to capture information from quotes that users and other processes generate through the course of the day. While writing policies for a customer, service agents often produce several quotes for the customer. For example, the customer wishes to see how the quoted premium changes with variations in levels of coverages or deductibles.

Without quote cloning, interim quotes are not available for capture during nightly processing for business intelligence (BI) needs. Quote cloning provides one approach for retaining some of this information long enough to capture it for BI needs. This feature also provides a mechanism for removing the clones from the PolicyCenter database after the information has been captured by an external system.

Quote cloning provides a framework for creating cloned copies of policy period quotes. Clones can include quotes that never result in a bound policy. Cloned quotes can be generated from submission, policy change, reinstatement, renewal, and rewrite policy transactions.

This framework is disabled in the base configuration.

In your implementation, you define these characteristics:

- Which quotes to clone. For example, you can process only submissions for new customers.
- How to process the cloned quotes. For example, you can configure quote cloning to process these clones as they are created. The processing extracts data from the cloned policy periods then saves that data to the external BI system. Alternately, you can create an integration that does an ETL extraction to the external BI system. This extraction might occur nightly.
- How often to purge processed clones. If you process the clones as they are created, you may choose to purge the clones every hour. If the processing occurs nightly, you may choose to purge the clones after the integration finishes the ETL extraction.
- Which processed clones to purge.

IMPORTANT Before enabling quote cloning, evaluate the impact on system performance, including the impact to memory and database.

Quote cloning does not have an end-user interface. It provides a framework that automated processes can use to generate BI data. Quote cloning is not intended to provide additional end-user functionality. The captured data does not appear in the PolicyCenter user interface.

If you need an end-user feature to create and compare multiple versions of a quote, PolicyCenter provides several options. For more information, see “Side-by-side quoting” on page 153, “Multi-version quoting” on page 161, and “Copying submission information” on page 93.

Quote cloning business example

When writing a new policies, an insurance company presents several quotes to a customer. The company wants to better understand which quotes result in bound policies. The company configures quote cloning to clone policy periods from submissions for new policies. Through configuration, quote cloning extracts relevant information from the cloned quotes and save the information to an external BI system. After processing, the cloned quotes are no longer needed. Quote cloning purges cloned quotes from the PolicyCenter database every 15 minutes.

Improving quoting and rating performance

In certain lines of business, generating quotes and rates on policies can take a noticeable amount of time. For example, this can occur on commercial policies with large numbers of coverables. To improve quoting and rating performance, consider using one or all of the following features:

- **Asynchronous quoting** – In certain lines of business, generating quotes for policies can take a noticeable amount of time. For example, this can occur on commercial policies with large numbers of coverables. Asynchronous quoting enables the quote to run in the background, so that the user can do work on other screens until the quote completes. Asynchronous quoting occurs when the number of coverables exceeds a threshold. In the base configuration, asynchronous quoting is available in the commercial lines of business: businessowners, commercial auto, commercial property, general liability, inland marine, and workers' compensation.
- **Parallel rating** – Parallel rating can improve the performance of generating quotes on policies with large numbers of coverables by rating coverables in parallel using multiple threads. The implementation requires Guidewire Rating Management. The base configuration implementations of parallel rating for the commercial property line of business. Two types of parallel rating are provided: parallel rating using entities and parallel rating using DTOs.
- **Parallel product model synchronization** – Parallel product model synchronization improves the performance of generating quotes by synchronizing the product model in parallel using multiple threads. Product model synchronization occurs at other times in the policy transaction so it can also improve performance outside the quote process. In the base configuration, parallel product model synchronization is enabled for commercial products.
- **Two-step quoting** – With two-step quoting, the first step validates the policy data and rates the policy, generating all cost and premium information, and bringing the policy period to Rated status. The second step completes post-rating tasks, such as generating forms, checking reinsurance, and raising underwriting issues. If the second step is successful, the policy period is then in Quoted status. With two-step quoting, you can delay or omit the second step entirely. For example, when working with multiple versions of a policy transaction, an underwriter does not need to generate forms, reinsurance, or underwriting issues for the policy until a specific version is chosen. Or in high volume quoting, only the actual price of the policy is relevant so it is unnecessary to do post-rating tasks. In the base configuration, two-step quoting is enabled for the commercial property line of business, but can easily be enabled for the remaining lines. Side-by-side quoting in personal auto uses two-step quoting. High volume quote requests use two-step quoting for all lines of business.

You can implement both asynchronous quoting and two-step quoting in the same policy line. If both are enabled, the first step, rating, is done asynchronously. The second step, finish quoting, is done synchronously.

Through configuration, asynchronous quoting and parallel rating can be modified to meet your needs and can be expanded to other lines of business.

See also

- *Configuration Guide*

Improving performance with parallel rating

Policies in certain lines of business have large numbers of coverables and generating quotes can take a noticeable amount of time. Parallel rating can reduce the time for generating quotes on these policies by rating coverables using multiple threads. In the base configuration, parallel rating is implemented in the commercial property line of business.

Note: Parallel rating requires Guidewire Rating Management.

You can use commercial property as an example for implementing parallel rating in other lines of business. Consider parallel rating for lines of business with policies that typically include a large number of coverables that do not share data.

Parallel rating works with other features of PolicyCenter, including asynchronous quoting and impact testing.

Commercial property rates the **Location** coverable in parallel. When parallel rating is enabled, any commercial property policy can be rated in parallel. Through configuration, you can specify the conditions required for rating a policy in parallel.

See also

- *Configuration Guide*

Parallel rating using entities

In the base configuration, parallel rating using entities is enabled by default for the commercial property line of business. Rate routines and code executed in parallel must be thread-safe. Enabling this type of parallel rating does not require changes to existing rate tables and parameter sets.

In the base configuration, the commercial property rate books, rate routines, and parameter sets in the small sample data set are compatible with parallel rating using entities.

Parallel rating using DTOs

Parallel rating using DTOs can reduce the time for rating even further than parallel rating using entities because of the following:

- DTOs lower the overhead related to bundle management, which occurs when accessing entities
- Read-only DTOs are inherently thread-safe

However, this type of rating is more complicated to implement. Enabling this type of parallel rating requires that rate tables, rate routines, and parameter sets access DTOs instead of entities. If you have entity-based rate tables, rate routines, and parameter sets, you must create DTO versions of these. This type of rating also requires that you create code for the DTOs that replace the entities in rating.

Sample data for parallel rating using DTOs

When parallel rating using DTOs is enabled, the small sample data set includes rate books, rate routines, and parameter sets for DTO rating of the commercial property line. These include **DTO** in their name or **dto** in their code. For example the sample data includes the **CP RTM Demo Rating DTO** rate book and the **BaseRate** rate table definition with the code **cp_coverage_base_rate(dto)**.

In Studio, the code that creates the sample data is in the **gw.rating.flow.scenario.CPDTORateflowScenario** class. This class creates the set of rate books and rate routines that uses DTO objects rather than entities in calculations and parameters. These rate books and rate routines provide the same functionality as the sample data for commercial property rating that does not use DTOs.

Improving performance with parallel product model synchronization

Parallel product model synchronization increases the speed of the synchronization by using multiple threads.

Because product model synchronization is a key piece of quoting, this can speed up the quoting process. In the base configuration, parallel product model synchronization is enabled for commercial products and occurs when the number of coverables exceeds a threshold.

In the base configuration default, each thread operates over a coverable and its clauses of a particular type. For example, one thread operates on the coverages on vehicle1, another on the conditions on vehicle1, another thread does the same for vehicle2. Through configuration, you can change both when parallel product model synchronization is triggered and the granularity of what each thread operates on.

See also

- *Configuration Guide*

Improving performance with asynchronous quoting

In certain lines of business, generating quotes for policies can take a noticeable amount of time. For example, this can occur on commercial policies with large numbers of coverables. Asynchronous quoting enables the quote to run in the background, so that the user can do work on other screens until the quote completes. Asynchronous quoting occurs when the number of coverables exceeds a threshold. In the base configuration, asynchronous quoting is available in the commercial lines of business: businessowners, commercial auto, commercial property, general liability, inland marine, and workers' compensation.

Asynchronous quoting is available in all policy transactions. Through configuration, asynchronous quoting can be modified to meet your needs and added to other lines of business.

To use asynchronous quoting, you must first enable it in the PolicyCenter installation.

Asynchronous quoting flow

In PolicyCenter, while working on a policy transaction, the flow for asynchronous quoting is as follows:

1. Navigate to a screen with a **Quote** button.

The screen has two identical **Quote** buttons, one each for synchronous and asynchronous quoting. Code in the PCF file controls which button appears. PolicyCenter displays the asynchronous **Quote** button if:

- Asynchronous quoting is enabled
- The policy line supports asynchronous quoting
- The number of coverables on the policy exceeds the threshold for asynchronous quoting

2. When you select **Quote**, PolicyCenter displays an alert that quoting will be processed in the background. To proceed, select **OK**.

a. PolicyCenter creates a work item for the quote and places it in a work queue for the Asynchronous Quoting work queue to process.

b. Until asynchronous quoting completes, PolicyCenter displays a message at the top of the policy transaction screens indicating that the quote is being processed. The policy transaction is locked and cannot be edited, and only the Back and Next buttons are available. You can view the current policy transaction and navigate to other screens and do work.

3. When quoting completes, PolicyCenter:

a. Unlocks the policy transaction.
b. Creates an activity that the quote completed. The activity subject indicates whether the quote completed successfully or completed with warnings or errors.

4. Navigate to **Desktop**→**My Activities**, and click the activity **Subject** to view the activity and the associated quote.

PolicyCenter displays the **Quote** screen for the policy transaction and the associated activity in the **Activity** tab. If the quote has warnings or errors, you can view them in the **Validation Results** tab next to the **Activity** tab. Other

more serious issues, such as rating or unexpected errors, appear on the policy transaction screen. A quote with only informational messages does not have a **Validation Results** tab.

[Asynchronous quoting and two-step quoting](#)

If both asynchronous quoting and two-step quoting are enabled for this policy line, you will see an asynchronous **Rate** button instead of **Quote**. The first step, **Rate**, is asynchronous. The second step, **Finish Quoting**, is synchronous.

Asynchronous quoting threshold

When asynchronous quoting is enabled, PolicyCenter performs asynchronous quoting if the current line of business supports asynchronous quoting and the policy exceeds the threshold.

In the base configuration, PolicyCenter triggers asynchronous quoting when the number of coverables exceeds the threshold of 30. The threshold is configurable. The following table shows the coverable that triggers asynchronous quoting in the base configuration.

Line of business	Coverable type
Commercial auto	Vehicles
Businessowners	Buildings
Commercial property	Buildings
General liability	Exposures
Inland marine	Signs
Workers' compensation	Covered employees

Asynchronous quoting for multi-version and side-by-side?

Asynchronous quoting and multi-version quoting are compatible and require no additional configuration. With multi-version quoting, each version is edited and quoted individually. If a particular version exceeds the threshold, it is quoted asynchronously.

Guidewire does not recommend implementing asynchronous quoting for side-by-side policy transactions. In side-by-side quoting, you view all versions on one screen and you quote all versions at the same time. Side-by-side quoting provides a condensed view of each policy that is not suitable for viewing policies with large numbers of coverables. In the base configuration, the code disables asynchronous quoting for side-by-side policy transactions.

Asynchronous quoting user interface

If asynchronous quoting is enabled, clicking the asynchronous **Quote** button triggers asynchronous quoting in policy transactions that meet the criteria for asynchronous quoting. Code in the PCF file determines whether the synchronous or asynchronous **Quote** button is visible. When quoting completes, PolicyCenter generates an activity.

[Asynchronous quoting and two-step quoting](#)

If both asynchronous quoting and two-step quoting are enabled, a **Rate** button triggers asynchronous rating. The second step, quoting (**Finish Quote**) is synchronous.

View asynchronous quoting activities

[About this task](#)

When asynchronous quoting completes, PolicyCenter generates an activity for the quote request.

Procedure

In the **My Activities** screen, click an asynchronous activity to view the activity in the policy transaction.

Create an asynchronous quote

Create an asynchronous quote in PolicyCenter using the sample data.

Before you begin

Change the asynchronous quoting threshold to 3 for commercial property in `gw/lob/common/AsyncQuoteCoverableThresholdByLine.gss`.

About this task

These instructions require that you run PolicyCenter in debug mode and install the small sample data set. Debug mode provides access to QuickJump commands that use the sample data.

Procedure

1. Log in as su.
2. In the **QuickJump** box, enter `Run Policy wDraft CP`.

PolicyCenter generates a commercial property submission and displays the **Policy Info** screen. Asynchronous quoting is triggered when there are more than three buildings.

3. Navigate to the **Buildings and Locations** screen and notice that among all policy locations there are more than three buildings.
4. Click **Quote**.

PolicyCenter displays a message that it will process the quote in the background and generate an activity when quoting completes.

Note: If two-step quoting is also enabled, a **Rate** button appears instead of **Quote**. Rating is done asynchronously. The second step displays a **Finish Quote** button.

5. Click **OK** to continue.

Until asynchronous quoting completes, PolicyCenter displays a message at the top of the policy transaction screens indicating that the quote is being processed. The policy transaction is locked and cannot be edited, and only the Back and Next buttons are available. You can view the current policy transaction and navigate to other screens and do work.

Wait for asynchronous quoting to complete.

6. Navigate to **Desktop**→**My Activities**, and click the activity **Subject** to view the activity and the associated quote. PolicyCenter displays the **Quote** screen for the policy transaction and the associated activity. If the quote has warnings or errors, you can view them in the **Validation Results** tab next to the **Activity** tab. Other more serious issues, such as rating or unexpected errors, appear on the policy transaction screen. A quote with only informational messages does not have a **Validation Results** tab.

Start asynchronous cancellation policy transactions

Before you begin

“Create an asynchronous quote” on page 427

About this task

The flow of an asynchronous cancellation is slightly different than a synchronous cancellation.

Procedure

1. Navigate to a policy that triggered asynchronous quoting.

2. Start a cancellation by selecting **Cancel Policy** from the **Actions** menu.

In synchronous quoting mode, the **Start Cancellation** button starts the cancellation, quotes, then displays the **Confirmation** screen. In asynchronous quoting mode, **Start Cancellation** takes you to the **Entry** screen instead of the **Confirmation** screen.

3. If this is an asynchronous quote, you must click **Quote** explicitly.

Improving performance with two-step quoting

With two-step quoting, the first step validates the policy data and rates the policy, generating all cost and premium information, and bringing the policy period to Rated status. The second step completes post-rating tasks, such as generating forms, checking reinsurance, and raising underwriting issues. If the second step is successful, the policy period is then in Quoted status. With two-step quoting, you can delay or omit the second step entirely. For example, when working with multiple versions of a policy transaction, an underwriter does not need to generate forms, reinsurance, or underwriting issues for the policy until a specific version is chosen. Or in high volume quoting, only the actual price of the policy is relevant so it is unnecessary to do post-rating tasks. In the base configuration, two-step quoting is enabled for the commercial property line of business, but can easily be enabled for the remaining lines. Side-by-side quoting in personal auto uses two-step quoting. High volume quote requests use two-step quoting for all lines of business.

Lines of business

In the base configuration, two-step quoting is enabled for the commercial property line of business. When two-step quoting is enabled in PolicyCenter, instead of a **Quote** button, a **Rate** button appears in policy transactions when the policy period is in Draft status. Click **Rate** to generate the premium for the policy and bring it to Rated status. After rating completes, execute post-rating tasks by clicking **Quote** to bring the policy to Quoted status.

You can try out two-step quoting by starting a commercial property submission.

Two-step quoting can easily be enabled in the other lines of business in the base configuration. In the base configuration, all lines of business are configured for two-step quoting.

Side-by-side quoting

When two-step quoting is enabled in side-by-side quoting, selecting **Versions**→**Side-by-Side** in a policy transaction brings all the generated versions to Rated status, displaying the premium for each. Selecting a version that is in Rated status executes the post-rating tasks for that version only, bringing it to Quoted status and taking you to the **Policy Review** screen for the selected version. If you make a change to any side-by-side version, that version returns to Draft status. If any version is in Draft, the **Side-by-Side Quoting** screen displays a **Rate All** button so that the new premium(s) can be generated, again bringing all versions to Rated status. Selecting a Draft version brings you to the **Policy Review** screen for that version with it still in Draft status. Depending on whether two-step quoting is enabled generally for the particular line-of-business, you can at that point either Rate or Quote the policy version.

High volume quote requests

High volume quote requests use the two-step quoting feature for all lines of business. When the quote-only instance generates a quote, only the first step is executed because all that is required at that point are the costs and premium for the policy. This brings the policy period to Rated status. If the quote is later brought into the PolicyCenter system of record (SOR), PolicyCenter performs the second step of two-step quoting, which executes the post-rating tasks. If successful, PolicyCenter raises the policy period to Quoted status.

One and two-step quoting workflows

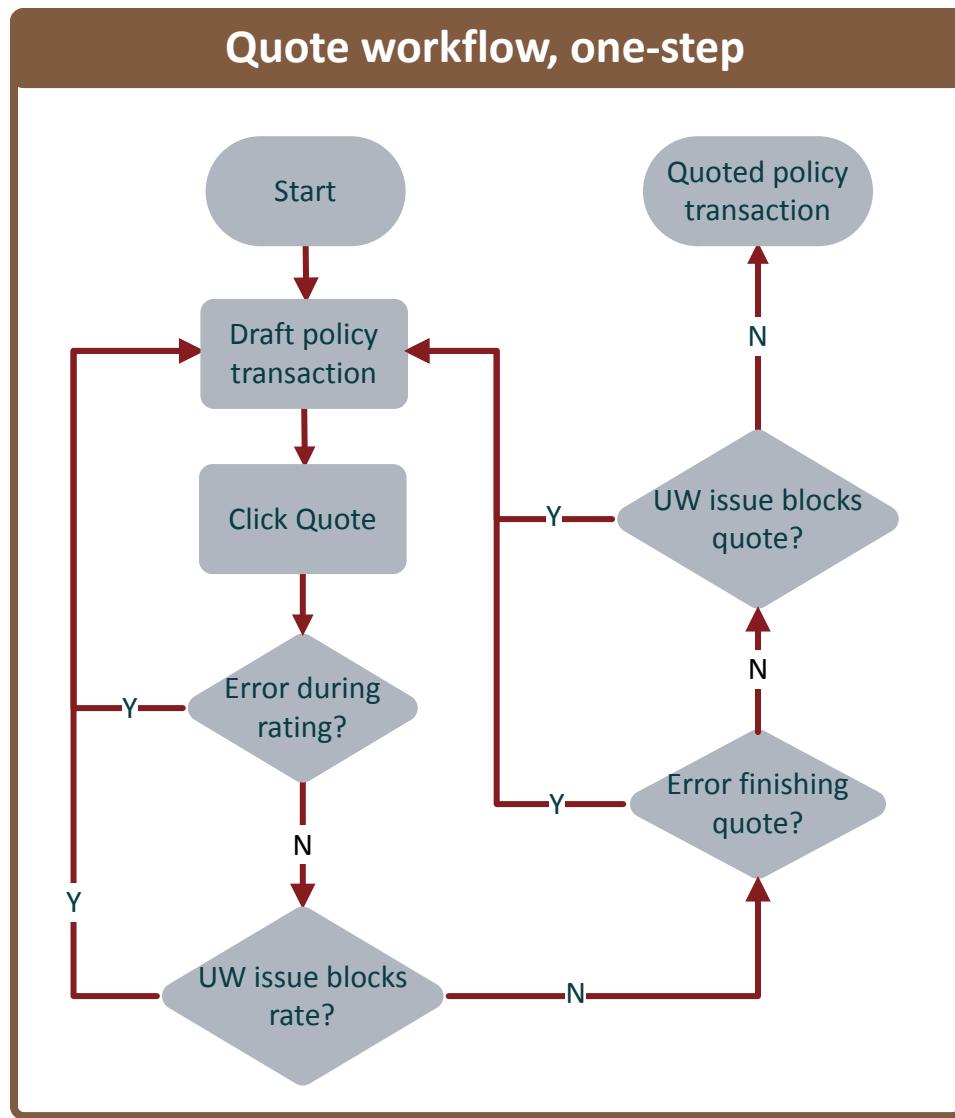
All PolicyCenter lines of business are configured to handle both one and two-step quoting. In the base configuration, two-step quoting is enabled only in commercial property, and the other lines of business use one-step quoting. You can easily enable two-step quoting in the other lines of business.

One-step quoting workflow

In one-step quoting workflow:

1. In PolicyCenter, create or edit a policy transaction.
2. After entering the policy data, click **Quote**.
 - a. PolicyCenter validates and rates the policy data, generating the costs and premium. If there are no errors, with the policy period in Rated status, the process continues without user intervention, generating forms, checking reinsurance, and raising underwriting issues. Without errors, the policy period is now in Quoted status.
 - b. You can now view the **Quote**, **Forms** and **Payment** screens in PolicyCenter.

The following illustration shows the workflow of a policy transaction using one-step quoting.



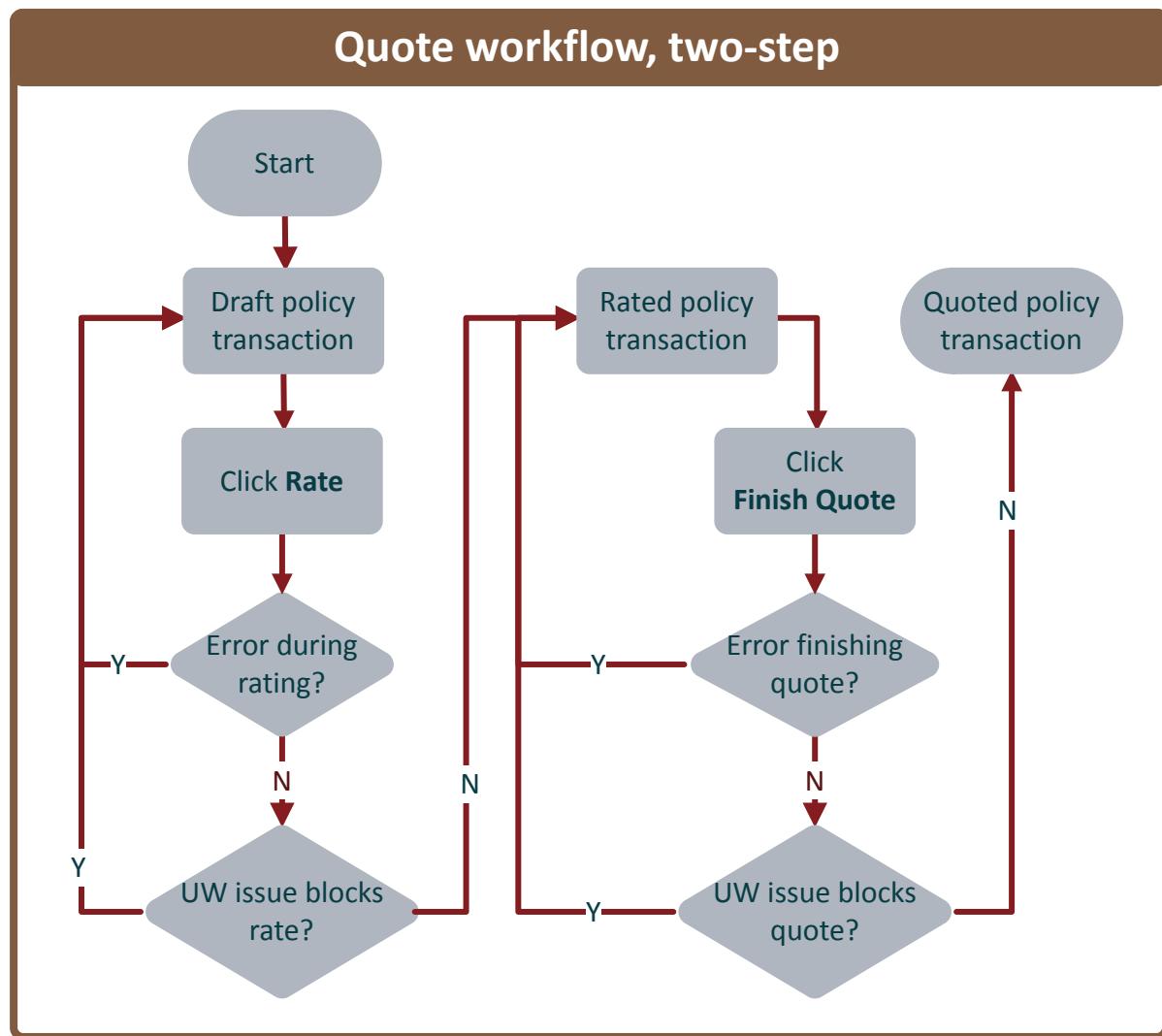
Two-step quoting workflow

In two-step quoting workflow:

1. In PolicyCenter, create or edit a policy transaction.
2. After entering data, click **Rate**. the policy data is validated and rated, generating the costs and premium.
 - a. If there are no errors, the policy period is in **Rated** status.
 - b. PolicyCenter displays the **Quote** screen with costs and premium information generated by the rating step.
3. If you like the rate and wish to proceed to generate a complete policy, click **Finish Quote**. This generates forms, checks reinsurance, and raises any applicable underwriting issues.

- a. If there are no errors, the policy period is in Quoted status.
- b. You can now view the **Quote**, **Forms**, and **Payment** screens in PolicyCenter.

The following illustration shows the workflow of a policy transaction using two-step quoting.



Risk assessment with Spotlight

PolicyCenter offers integration with Guidewire Spotlight. This integration provides a way to exchange data between PolicyCenter and Spotlight. This integration allows PolicyCenter to display and utilize results from location-specific risk assessments evaluated in Spotlight, as well as the inclusion of a map of the location.

Note: For information about obtaining access to Spotlight, contact your Guidewire support representative.

Risk assessment with Spotlight overview

In the base configuration, the commercial property line of business provides access to Spotlight and enables the risk assessment services. On the **Location Information** screen, Spotlight provides an embedded map for a commercial property location. From this screen, you can open Spotlight to the latitude and longitude. If latitude and longitude are not provided, Spotlight uses the address to locate the pin. In Spotlight, you can change the location by moving the pin. Spotlight returns location-specific risk evaluation results for all assessments included in the selected risk profile. The **Policy Location** screen displays this data in a table regardless of which specific assessments are configured for parsing.

Flood risk information is configured as an example of how to parse out specific assessment results from the data Spotlight returns. Flood risk is not automatically returned when Spotlight is enabled. For more information, see the *Configuration Guide*.

On the **Risk Analysis** screen, you can update risk evaluations.

Spotlight Integration Requires HTTPS

Communication with Guidewire Spotlight must be secured by serving PolicyCenter over HTTPS. Failure to do so may result in various browser warnings as well as potentially transmitting sensitive information over insecure channels. Therefore, implement HTTPS and disable HTTP on PolicyCenter servers.

For Spotlight services, in particular the interactive integration, to function properly and securely, the PolicyCenter application server must be run using HTTPS with a valid certificate.

Risk assessment user interface

In the base configuration, Spotlight interactive and risk assessment services are provided in the commercial property line of business.

Note: If you run PolicyCenter using HTTP during development, the web browser may be unable to complete loading Spotlight. You may need to enable loading scripts from unauthenticated sources in your browser.

For information on enabling Spotlight services, see the *Configuration Guide*.

Using the risk assessment Location Information screen

The right side of the **Location Information** screen displays data returned from Spotlight. This data includes the time of the last evaluation, latitude and longitude, risk profile data, assessment results, and an embedded map.

When you enter the **Location Information** screen, PolicyCenter determines whether the current location data differs from the data used to generate the existing location risk assessment, if one exists. If there is a difference, the screen displays a warning indicating that the current risk assessment may be out of date. This warning also appears on the **Buildings and Locations** screen for any location with a stale risk assessment.

Changing the pin in Spotlight updates **Latitude** and **Longitude**, and may impact the assessment results returned for an evaluation on the right of the **Location Information** screen in PolicyCenter. The policy location **Address** and other fields on the left of the screen are not affected by changes in Spotlight.

Clicking **Update Map** updates the map with the current latitude or longitude, or current address if those values do not exist. If you change the location address and wish the map to reflect that change, click **Evaluate in Spotlight**. In the Spotlight text box at the top left of the screen, replace the latitude and longitude with the new location address. You can adjust the pin location, then return to PolicyCenter.

Perform a risk assessment

Procedure

1. Click **Evaluate in Spotlight** to navigate to Spotlight.

You can evaluate the risk in the Spotlight application.
2. Move the latitude and longitude of the pin that represents the location, or replace the latitude and longitude with the new location address.
3. Click **Update Map** to update the map with the location information.
4. From the drop-down list, select a risk assessment profile.
5. Optionally, click **Evaluate** to view the new risk data in Spotlight.

Spotlight returns new risk data whether or not you click **Evaluate**.
6. Click **PolicyCenter** to return to PolicyCenter.
7. Select **Spotlight returned new data**→**Use this new data** to save the selected risk profile, latitude and longitude, and new risk assessment in PolicyCenter.
8. Click **OK** to update the risk assessment data.

Beneath the map, the screen displays fields related to the last Spotlight evaluation, latitude and longitude, and assessment results.
9. In **Risk Profile**, click **Show** to view complete risk assessment data returned for the specified risk profile.

See also

- *Configuration Guide*

Update risk evaluations

The **Update Risk Evaluations** button appears on the **Risk Analysis** screen of a Commercial Policy policy transaction. Updating risk evaluations creates activities for both success and failure.

The button is visible only on unbound jobs and only if Spotlight is enabled for multiple location risk assessment. It is available only when the policy is writable. The button becomes unavailable when pressed and is unavailable when the policy period has a pending request. Additionally, the screen displays an alert when the policy period has a

pending request. When the risk assessment completes, you must navigate to another screen to refresh the **Update Risk Evaluations** button and alert.

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.

Use policy forms for the information that comprises the policy contract. For generating and tracking information that is not part of the policy contract, use documents. See “Document management” on page 677.

Policy 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. After quoting, 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 (or after the policy is issued in a submission), 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 or issuance. When the policy is bound or issued, your integration code sends XML data describing the form to an external system which prints the forms to paper or electronic format.

The forms feature of PolicyCenter has the following components:

What	Where to configure	Description
Forms basic definition	In PolicyCenter, go to Administration → Business Settings → Policy Form Patterns .	View or add policy form patterns 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 745 for more information about how to administer forms.
- The *Integration Guide* for more information about inference classes and forms printing integration.
- The *Integration Guide* for more information about document management.

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.

See also

- “Importing and exporting policy data spreadsheets” on page 760
- *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.

Export buildings and locations to a spreadsheet

About this task

To export buildings and locations from a policy in PolicyCenter to a spreadsheet:

Procedure

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.

Export	<ul style="list-style-type: none">• Commercial Property Locations – Exports a spreadsheet that enables you to add new locations.• Commercial Property Buildings – Exports a spreadsheet that enables you to add buildings. If needed, you can also specify new locations for the new buildings. <p>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.</p>
All from this version	Exports a spreadsheet that contains all existing coverables, enabling you to make changes to existing policy data and add new coverables. You can only import this spreadsheet into the same policy transaction from which it was exported.
Template for any policy transaction	Exports a template spreadsheet that contains only column headings, enabling you to add new coverables only. You can use this spreadsheet to import new policy data into any policy transaction.
Format	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 All Available . To export a subset of fields, select the corresponding format from this list. Consult the person who designed the export formats to determine the appropriate formats to use for various situations.
Language	Lists available languages. Select the language that appears in the exported spreadsheet column headings.
Locale	Lists available locales.

4. Click **Export to Spreadsheet** to export the specified spreadsheet. Specify a location and file name for the spreadsheet. PolicyCenter provides recommendations for file names 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 exported 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 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.

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.

Add buildings to an existing location in the spreadsheet

Procedure

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.

To add multiple buildings to the same location, paste the same set of location data to as many rows as needed first. Then fill in the remaining building data for each building. If multiple buildings share duplicate values, copy and paste those cells between rows. Remember that you can copy and paste only cell ranges, not entire rows.

8. Save and import the spreadsheet.

Add buildings to new locations in the spreadsheet

Procedure

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.

Add multiple buildings to the same new location in the spreadsheet

Procedure

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.

Make changes to buildings or locations in a policy change

About this task

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 in a policy change:

Procedure

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 “Make changes to buildings or locations in a policy change” on page 442 for each building or location that must be changed.
4. Save and import the spreadsheet to PolicyCenter.

Import spreadsheet into policy transaction

Procedure

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 exported spreadsheets” on page 440.

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 insurer. 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 474.

- 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.

See also

- 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 topics describe the most important product model patterns:

In addition to these main template patterns, there are also form patterns and modifier patterns. For more information, see 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

- The *Product Model Guide* for a description of the **Product** screen interface

Policy line 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

- The *Product Model Guide* for a description of the **Policy Line** screen

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 coverage term 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 in the PolicyCenter interface.
- The **Removal script** Gosu script, which runs when a user removes a coverage in the PolicyCenter interface.

See also

- *Product Model Guide*

Product model 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. A condition is policy provision that defines required behaviors for the carrier and insured. A common example of a condition is a state amendatory endorsement that details requirements in a single jurisdiction.

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.

Product model 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.

Coverage term 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

- *Product Model Guide*

Product model 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:

- | | |
|------------------------------|------------------|
| • Availability lookup tables | • Offerings |
| • Availability scripts | • Reference Date |

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

- *Product Model Guide*

Determining the reference date for availability

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 insurer change over time, as do the rates that are applied to these. These changes occur because insurers 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, an insurer 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

- *Product Model Guide*

Making a product model pattern available by policy transaction type

You can specify whether to use a pattern based on the policy transaction type. Insurers may want to make a coverage or other pattern available to new business before allowing for renewal business. Insurers 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

- *Product Model Guide*

Grandfathering in the product model

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 be automatically removed from existing usages.

Grandfathering is available on:

- | | |
|--|--|
| <ul style="list-style-type: none">• Coverages• Exclusions• Conditions• Coverage terms | <ul style="list-style-type: none">• 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 insurer decides to offer a particular coverage in California until December 2017, then grandfathered the coverage until December 2019. The insurer also offers this coverage in Nevada until October 2017, then grandfathered the coverage until May 2020.

See also

- *Product Model Guide*

Reloading availability in PolicyCenter

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

- *Product Model Guide*

Offerings in the product model

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.

An insurer 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.

Filtering the product model in availability

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 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

- *Product Model Guide*

Select and change an offering

About this task

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 “Create a submission” on page 95.

Procedure

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 Line** screen. Because you selected the **Partners** offering, the **Policywide Property Deductible** is optional. 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>**.
8. Click **Businessowners Line** in the left side bar to return to that screen. Notice that the **Policywide Property Deductible** is now required (it cannot be deselected).

Schedules overview

Schedules are lists that contain detailed information about an insured’s coverables and coverages. There are various types of schedules.

Schedules as lists of coverables

Some schedules are lists of fundamental and more complex coverables, such as the list of vehicles in a commercial auto or personal auto policy. In PolicyCenter, each item in this type of schedule corresponds to a dedicated entity instance, and can specify properties for that entity instance. In commercial auto the entity is **BusinessVehicle**, and in personal auto the entity is **PersonalVehicle**. You can use the examples in commercial auto or personal auto as examples for implementing schedules that are lists of fundamental coverables.

Generic schedules for clauses

Some types of schedules capture detailed information about clauses (coverages, conditions, and exclusions) attached to the coverable. For these types of schedules, PolicyCenter provides *generic schedules*. *Scheduled clauses* are clauses with a generic schedule attached.

In the base configuration, the Homeowners line of business uses generic schedules. A clause that uses a generic schedule is the scheduled personal property coverage in Homeowners.

Using generic schedules, you can implement the following types of schedules:

Schedules

Capture information per scheduled item, such as a name or description. In general, schedules of this type are not used directly in rating, but are often taken into consideration during underwriting and usually are included in forms.

Schedules with terms

Capture information per scheduled item. This can be any type of clause, but is usually a coverage. Each scheduled item includes the coverage terms from one coverage. The coverage term options selected for each scheduled item are passed to the rating engine and potentially affect the cost of the policy.

See also

- *Product Model Guide*

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. 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.

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	How PolicyCenter uses this dimension
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.
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.

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:

Model date

March 1

Effective date

April 1, effective until the end of the period

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.

Model time examples

The following are examples of model time:

- 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.
- 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.

Effective time examples

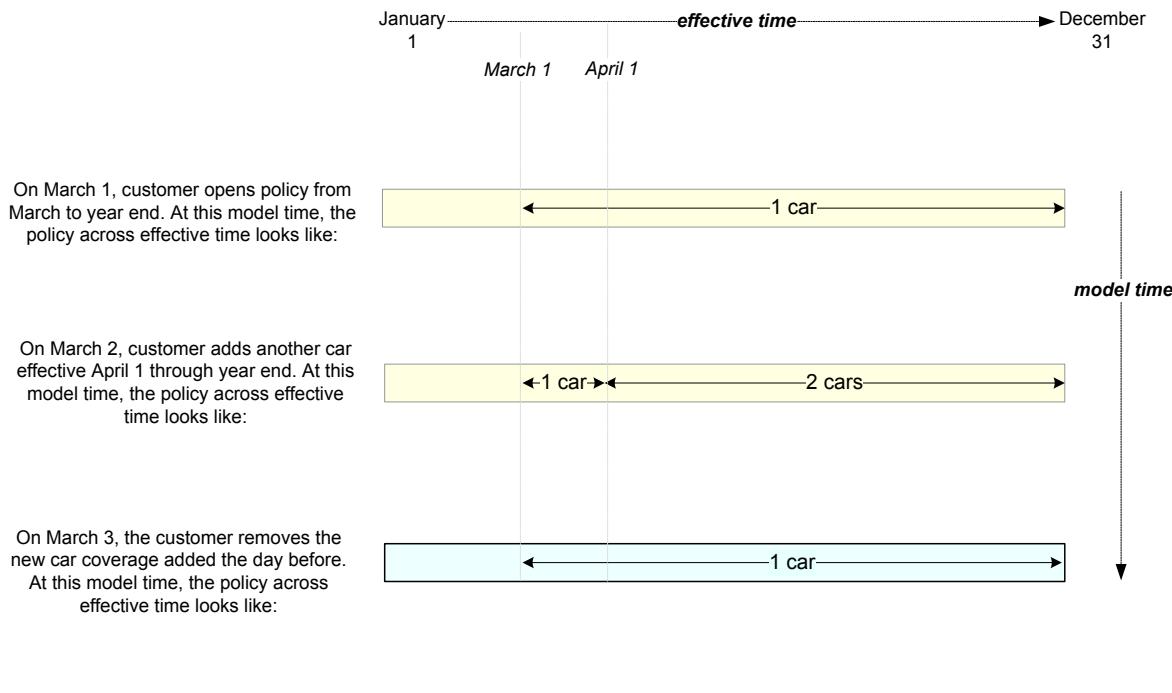
The following are examples of effective time:

- 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.
- 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.
- A change to an effective date in the past: canceling a policy effective last month.

Policy in model time versus effective time

The following diagram shows a policy across model time and effective 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

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 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 459.

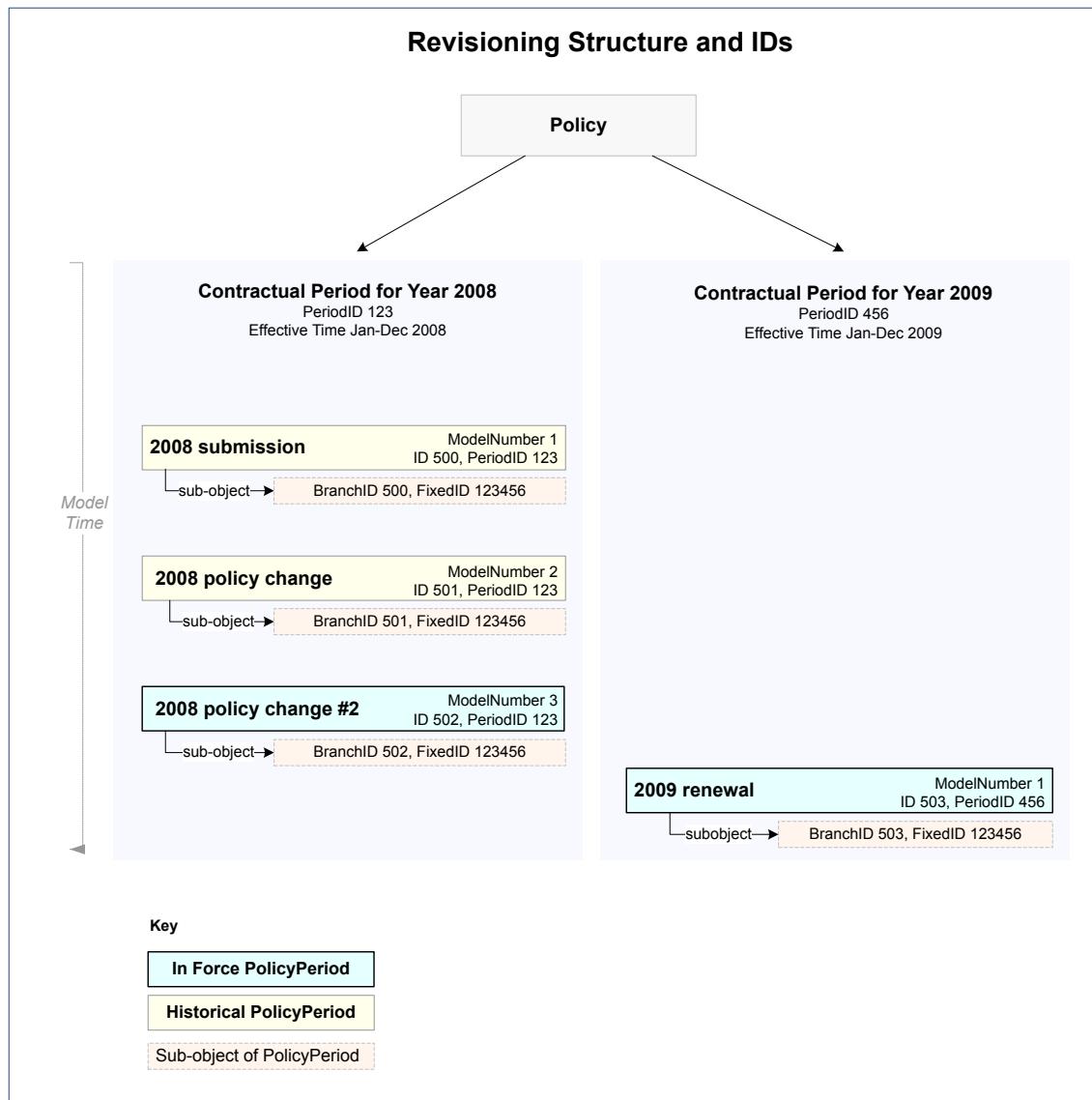
For these reasons, PolicyCenter knows which rows represent the same object because they share an ID called a *fixed ID*, stored in its `FixedID` property. If the fixed IDs for two vehicles match, they are versions of the same vehicle, not two different vehicles. If the fixed IDs do not match, they represent 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.



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 revisioned graph, it does not behave like other revisioned 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.

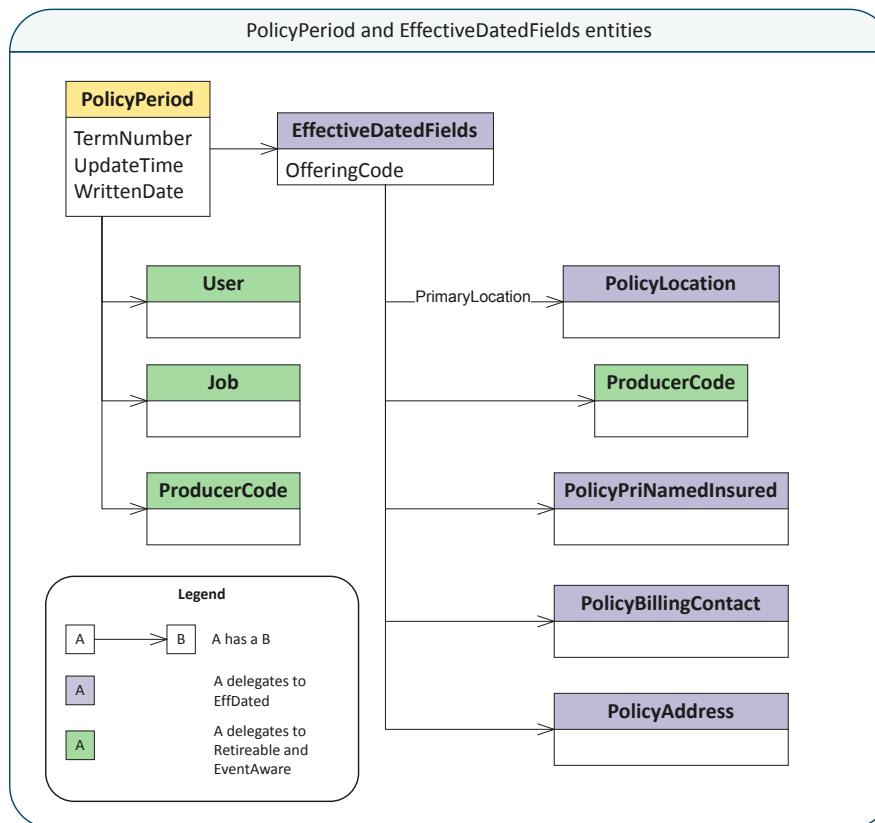
For example, the **PolicyPeriod** has unrevised properties that apply to the full term. Some of these properties are:

- **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 1/1/2019, the array contains one

`PolicyContactRole` object. On 3/1/2019, a policy change adds two `PolicyContactRole` objects. On 5/1/2019, 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 1/1/2019 to 1/1/2020. 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 1/1/2019 to 1/1/2020 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 7/1/2019 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 1/1/2019 to 7/1/2019, and then it is `Beta` from 7/1/2019 to 1/1/2020.

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 it was `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 January 1 of the next year

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.

Subobjects 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

↓
Subobject

VEHICLE

Color: RED
EffectiveDate: null
ExpirationDate: null
FixedID: 12345

A PolicyPeriod entity's subobject 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

VEHICLE

Color: BLUE
EffectiveDate: August 1, 2008
ExpirationDate: null
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: RED
EffectiveDate: null
ExpirationDate: August 1, 2008
FixedID: 12345

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

Subobject 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 **MostRecentModel**.

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 464.

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 472, “Preempted jobs” on page 474, and “Applying changes to future renewals” on page 477.

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 462.

In general, you get a version list from a revised object by getting its `VersionList` property. The result has the type specific to that object with the `VersionList` suffix, for example a `Building` version list has type `BuildingVersionList`. In the unusual case that you write general purpose code that operates on multiple revised entity types, declare variables as the base type `EffDated` or `EffDatedBase`. For those types, instead of getting the `VersionList` property, get the property `VersionListUntyped`.

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 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.

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 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 return returns a list. For details, see the *Gosu Reference Guide*.

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 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

The version list API provides methods to access to all versions of an effective dated object across all dates in effective time for the duration of a policy period.

Version list API methods that query an effective dated object

The following methods on the version list act upon an effective dated object (entity instance). Every version list has properties and methods that are common to all version lists. The version list also has properties that mirror property names on the source object.

The version list name includes the entity instance for which it is a version list appended with the suffix `VersionList`. For example, a version list for a `PersonalAutoLine` entity instance has type `PersonalAutoLineVersionList`.

Examples assume an entity instance of type `PersonalVehicle` with a property called `Drivers`. Each `Driver` is of the type `PolicyDriver`.

AllVersions

Gets all versions of this entity instance in this period.

Returns a list of entity instances of the entity type. Each entity instance has a unique effective date range that does not overlap. For `PersonalVehicle`, this property returns `List<PersonalVehicle>`.

This property gets all versions of this entity instance across effective time in this policy period. Each entity instance reference is set to edit in window mode.

This property can access properties on an entity instance where the properties are not array properties. You use this to iterate across all versions and get the desired property from each one.

You can use this method in rating code to iterate across all effective time versions of an object that you need to send to the rating engine.

Note: When getting this property on a version list for a `PolicyPeriod` (the graph root), this property contains only one version since this root entity instance is not revised.

asOf(date)

Gets the version of this entity instance that is effective at the specified date, if such version exists.

Returns one entity instance as of the specified date, if such instance exists. The entity instance is in window mode. If the entity instance does not exist on the specified date, returns `null`. This can occur if the entity instance has been removed, canceled, or not yet added to the policy. For `PersonalVehicle`, this property returns a `PersonalVehicle`.

Use this method 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.

`hasGaps`

Check if an entity instance has effective-date gaps.

Returns `boolean`.

If `true`, the entity instance has at least some amount of non-effective time between two ranges of effective time in that contractual period.

`hasOverlaps`

Check if an entity instance has overlapping duplicates across effective time.

Returns `boolean`.

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 before binding the branch. You can choose to use this method in your own validation code or other Gosu code.

`getAllVersionsUntyped`

Get all versions of this object, but typed to the root of all revisioned entities.

Returns a list of effective dated entities: `List<EffDatedBean>`. The type for each item is the root class of all revisioned entities, which is `EffDatedBean`. Typically, you need to cast each item to the specific entity subtype.

This method is similar to the `AllVersions` property, but with a different return type declaration.

`getVersionAsOf(date)`

Gets the entity instance as of a date, but typed to the root of all revisioned entities, `EffDatedBean`.

Returns an entity instance typed as the root of all revisioned entities (`EffDatedBean`).

This method is the same as the `asOf` method, but with a slightly different return type.

Version list API methods that query an array of entities

The following are methods on the version list that query an array of entities. 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.

Examples assume an entity instance of type `PersonalVehicle` with a property called `Drivers`. Each `Driver` is of the type `PolicyDriver`.

`property`

Use `property` where `property` is the name of the property in source entity instance returns a list of all version lists for this property. The version lists are typed to the property type on the source entity with a suffix of `VersionList`. Each version list in the result represents the source entity instance and all its versions for this period. The source entity instance and its versions have the same fixed ID.

There is a method for each property on the source entity instance that contains an array. For `PersonalVehicle`, the `Drivers` property returns `List<VehicleDriverVersionList>`.

`propertyAsOf(date)`

There is an `AsOf(date)` method for each property on the original entity instance that contains an array.

This method gets the contents of this property as of a particular date. Add the `AsOf` suffix to the name of the property in the source entity instance. This is a method even though its first character is capitalized. Pass the date as an argument.

For `PersonalVehicle.Drivers`, the `DriversAsOf` method returns `List<VehicleDriver>`.

Returns a list, not an array, of the type contained in the array property. This could be an empty list if no child entity instances in that array are effective at that date due to being removed or canceled. The return value can be an empty array. This method does not return `null` nor throw an exception.

`addToProperty(obj)`

There is an `addTo` method for each property on the original entity instance that contains an array.

Add an entity instance to the version list that represents the property that is array of entity instances of that type.

This method returns `void`.

The method is `addTo` appended with the source property name. This method takes an entity instance of the type of the original property (in this example, `VehicleDriver`). This change always happens in window mode to preserve the effective and expiration dates on the entity instance. In contrast, when adding entities in slice mode, PolicyCenter overrides the effective date with the slice date.

Version list API methods that are not type-safe

The following version list methods use *type reflection* (dynamic access) and are rare in typical code. In general, avoid using these methods because reflection APIs prevent Gosu from checking the correctness of arguments at compile time.

IMPORTANT Use these methods only for algorithms that are impossible with type-safe APIs. This can occur if you do not know property names at compile time.

These methods are similar to `property` in “Version list API methods that query an array of entities” on page 468. However, these methods uses type system reflection to get the property. Pass the property as an argument.

`getArray(propName)`

Get all version lists for this property using reflection to specify the property name.

Returns 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.

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.IArray PropertyInfo  
var driversArray = vehicleVL.getArray(driversProp)
```

`getArrayAsOf(propName, date)`

Get a list of the objects in the array that are effective at the particular date using reflection to specify the property name.

Returns 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. For example, cast the version list to a more specific subclass such as `List<VehicleDriverVersionList>`.

Returns an empty list if there no child objects in the 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 method does not return null and does not throw an exception for this condition.

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 the *Gosu Reference Guide*. For more information about blocks, see the *Gosu Reference Guide*.

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 vehicle:

```
var firstVersionUnsliced = vehicleVL.AllVersions.First()
var origDrivers = firstVersion.getSlice(firstVersion.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 vehicle's garage location at a specific date

Suppose you want to get a vehicle'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 466 for related discussion.

Get all available drivers for a vehicle 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 results in 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 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 472.

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**→**Out-of-Sequence Conflicts** tab 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 out-of-sequence conflicts, the **Policy Review** page for the job contains an **Out-of Sequence** tab. PolicyCenter alerts you to out-of-sequence conflicts when you try to bind the job.

Create an out-of-sequence policy transaction

About this task

This task uses the Ray Newton personal auto policy from the small sample data set. The policy is in force, and the policy number is 3316157326.

Procedure

1. Open any personal auto policy that is currently active. Note the dates of the period. If necessary, create a new policy that is effective right now and extends a few months in the future.
2. Select **Actions**→**Change Policy**.
3. In the **Start Policy Change** screen, enter an effective date in the middle of the period (2 months from the start date), and a short description. Click **Next**.
4. In the left sidebar, click **PA Coverages**.
5. In **Uninsured Motorist - Bodily Injury**, change **Uninsured Motorist - BI Limits** to **250/500**. Click **Quote**.
6. Click **Issue Policy**.

7. On the **Policy Change Bound** screen, click **View your policy**.

Create the out-of-sequence policy change

8. Navigate to the policy and select **Actions→Change Policy**.
9. Enter an **Effective Date** that is in the period but before the effective date of the last change you made (1 month from the start date). Enter **out-of-sequence** in the **Description**. Click **Next**.

A dialog appears warning you of the out-of-sequence transaction. click **OK** to continue.

10. In the left sidebar, click **PA Coverages**.
11. In **Liability - Bodily Injury and Property Damage**, change the value of **Auto Liability Package** to **15/30/5**. This is the out-of-sequence change.
12. Click **Quote**.
PolicyCenter warns you that there are out-of-sequence conflicts that must be resolved prior to quoting.
13. As suggested, go to the **Policy Review** screen, and select the **Change Conflicts** tab.
This screen displays all conflicts and lets you decide whether to override the future conflict.
14. Choose your override method.
 - Override all conflicts or none by using the **Override All** or **Override None** buttons.
 - Override (merge) individual conflicts with your recent change by selecting **Yes** in the **Override Future Conflicts** column. Overriding later-effective-date jobs has the effect of merging forward your change for the rest of the contractual policy period.
15. Click **Submit** to finish overriding.
16. Quote and bind the policy 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 lifecycle. Sometimes this delay is due to technical reasons such as contacting external systems. 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 displays a warning to the user.

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 **Handle Preemption** to handle the preemption, PolicyCenter displays a tree showing the preemptions.

You can use the tree navigation (clicking on - and + signs) to hide or show parts of items in the hierarchy.

Use the buttons 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.

[Effective date later](#)

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.

[Effective date earlier](#)

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 472.

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. This can happen if you make a change to a vehicle that has been removed in a preempting branch. Because of the preempting branch, there is no longer a vehicle in the newly merged branch. If such rare cases occur, after PolicyCenter reapplyes changes, PolicyCenter opens a worksheet to notify you about change conflicts. This is just a notification. It requires no action.

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 the *Integration Guide*.

Create a preempted job

Procedure

1. Open a policy that is currently active. Notice the dates of the policy period. If necessary, create a new policy that is effective right now and extends a few months in the future.

This example uses a personal auto policy.

2. Select **Actions→Change Policy**.

3. Enter an effective date in the middle of the policy period and a short description.

For example, select an **Effective Date** two months from the original effective date. Enter **Change 1** in the **Description**.

4. Make a change to a coverage.

For example, select **PA Coverages** and change **Uninsured Motorist - Bodily Injury**→**Uninsured Motorist - BI limits to 250/500**.

5. Click **Quote** to quote the policy change.

6. In the Info Bar, click the policy number to go to the **Policy Summary** screen.

The policy change is not completed. It has been quoted but not bound.

7. Select **Actions→Change Policy**.

PolicyCenter displays a warning that there is another open policy change and that you might want to wait.

8. Enter an effective date later than the first policy change but still in the policy period. Add a short description. This will be the preempted policy change.

For example, select an **Effective Date** three months from the original effective date. Enter **Change 2** in the **Description**.

Note:

If you select a date is between the original effective date and the start of the first policy change, the preemption will also be out-of-sequence.

9. Click **Next**.

10. In the Info Bar, click the policy number to go to the **Policy Summary** screen.

11. In **Pending Policy Transactions**, click to open the first policy change, with **Status of Quoted**.

12. Click the **Transaction #** to open the quoted policy change.

13. Click **Issue Policy**.

The **Policy Change Bound** screen displays a link **Your policy change preempted Policy Change...**. This preempted policy change is the second policy change.

14. Click the **Your policy change preempted Policy Change...** link to jump to the second policy change.

15. The screen displays a message that this policy change was preempted and that you need to handle preemptions before continuing.

16. Click **Handle Preemption or Withdraw Transaction**.

If you choose **Handle Preemption**, you have three choices:

- **Apply All Changes** from the first policy change to the current (second) policy change.

- **Withdraw** the current (second) policy change.

- **Decide Later** how to handle the preemptions. You cannot quote the policy change until you handle preemptions.

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 481
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 462.
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 462.
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 472.
preemption	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 474. A preempted job can also contain out-of-sequence changes. You must handle both issues before binding the job.
merge changes	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. Contrast with the term <i>apply changes</i> . See “Details of merging and applying changes” on page 483.
apply changes	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 re-create 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 483.

For a full reference of revisioning properties on the `Policy`, `PolicyPeriod`, and `PolicyPeriod` subobjects, see the next section, “Revisioning properties reference” on page 480.

Revisioning properties reference

This topic includes revisioning properties reference information.

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 revised subobjects.

The following table lists important revisioning properties on a `Policy` entity instance:

Property	Type	Description
Periods	<code>PolicyPeriod[]</code>	An array of all <code>PolicyPeriod</code> entities associated with this policy including: <ul style="list-style-type: none"> • All contractual policy periods (including renewals, both bound and unbound) • All bound enforced branches • All bound historical (superseded) branches • All draft branches You typically do not access this property directly. This property contains much data that must be filtered in typical use. Instead, use the <code>BoundPeriods</code> property.
BoundPeriods	<code>PolicyPeriod[]</code>	An array of all bound <code>PolicyPeriod</code> entities associated with this policy including: <ul style="list-style-type: none"> • All contractual policy periods • All bound branches, even superseded ones 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.

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 revised 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 462. This is a read-only property. To get this <code>PolicyPeriod</code> at a different slice date, use the <code>getAsOf</code> method, described further in “Slice mode APIs” on page 463.
		Note: PolicyCenter never persists the slice date value itself in the database. The slice date property exists as a special property on the in-memory entity instance that Gosu can access.
Slice	Boolean	If true, this <code>PolicyPeriod</code> 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 <code>PolicyPeriod</code> at a different slice date, use the <code>getAsOf</code> method, described further in “Slice mode APIs” on page 463.

Property	Type	Meaning
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 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"> PolicyCenter sets this to false on the previous branch. PolicyCenter sets this to true on the newest branch in the same database transaction. 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.
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 474. 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. Note: 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 481
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 revisioned 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 properties related to revisioning 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	<p>Identifies a single object across contractual policy periods, both:</p> <ul style="list-style-type: none"> • Within the same PolicyPeriod but with different effective dates • Across multiple PolicyPeriod entities in one contractual policy period <p>This value must be non-null.</p>
BranchValue	Integer	<p>Note: At the database layer, BranchValue is the BranchID column. 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>Note:</p> <p>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 462.

Property	Type	Meaning
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 474 and “Details of merging and applying changes” on page 483. 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 472.
- **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 474 and “Applying changes to future renewals” on page 477.

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 453.

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 477.

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: <ul style="list-style-type: none"> Only if the changed entity instance exists on the preemption branch at the effective date of the change. For a non-slice edit: <ul style="list-style-type: none"> Only if the changed entity instance exists on the preemption 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.
DiffWindow	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 453.

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 472 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 474 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.

Out-of-sequence change	Later effective date slice changes	Merged result
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.
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 Change 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 Change 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 the *Integration Guide*.

Multicurrency features

Through multicurrency, PolicyCenter provides the ability to write policies that provide insurance for assets in different currencies. For example, an insurer offers a commercial property policy that can include properties in more than one country. With multicurrency, the policy values the properties in the currency of their locations.

See also

- “Multicurrency integration between BillingCenter and PolicyCenter” on page 780
- *Configuration 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.

Single currency and multicurrency in PolicyCenter

In the base configuration, multicurrency features in PolicyCenter enable you to create multicurrency lines of business. PolicyCenter operates in both single and multiple currency display modes. Whether you run PolicyCenter with a single currency or with multiple currencies, PolicyCenter objects and user interface elements for displaying monetary amounts include both an amount and a currency property. Objects, such as accounts and contacts, contain properties for specifying currency properties, such as a preferred currency. Producer organizations can have billing plans and producer codes can have commission plans in different currencies.

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 visible only in the base configuration when you enable multicurrency display mode. Even when multicurrency display is set to single, currency fields still are 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.

Multicurrency terminology

The following terms are associated with multicurrency.

Term	Description
Coverage currency	The currency for a particular coverage term in the policy contract.

Term	Description
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 insurers 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 objects 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 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, an insurer in the United States writes homeowners policies to insure assets in North America. One customer has properties in the United States and Canada. The insurer 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 insurers with multinational clients need to value coverage terms in more than one currency within a single policy. For example, an insurer 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

- *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 insurer 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 insurer will implement their own exchange rate service.

See also

- *Configuration Guide*

Multicurrency and rating

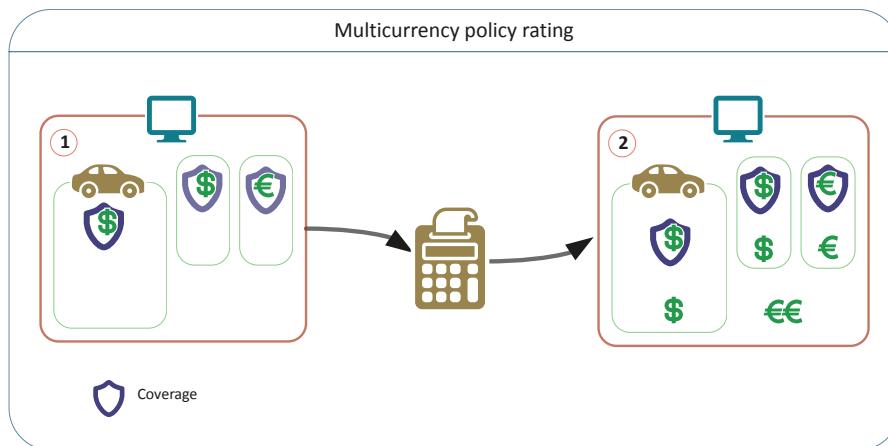
In the base configuration, the rating engine creates cost data in the currency of the associated coverage. For cost data not associated with a coverage, including certain taxes, the rating engine uses the preferred settlement currency on the policy period. The rating engine returns the cost data to PolicyCenter. 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.

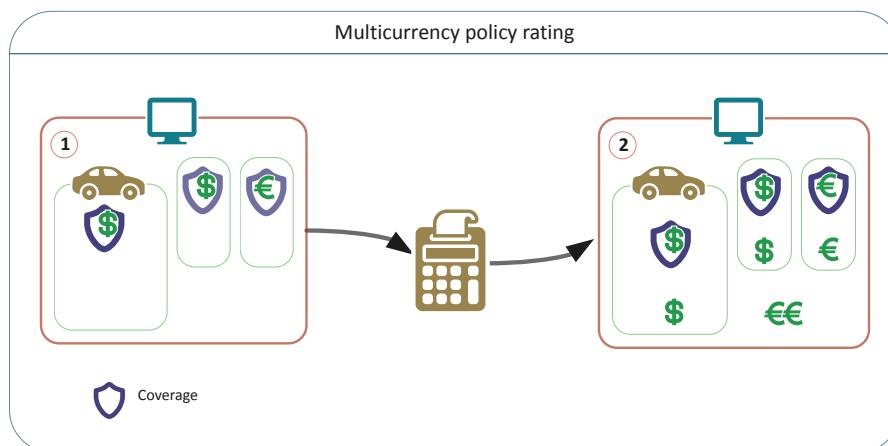
Example

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

- “Calculating transactions” on page 406
- *Configuration Guide*

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

- *Configuration Guide*

Rating monetary inputs not modeled as coverage terms

In PolicyCenter, the default system for rating and the rating provided by Guidewire Rating Management handle multicurrency monetary inputs which are modeled as coverage terms.

For monetary inputs to rating which are not modeled as coverage terms, your code must ensure that these rating inputs are in a currency that the rating system can handle. You may need to add validations for these inputs. Your validations may need to handle cases where the coverage currencies changes during the life of the policy.

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 program that covers jurisdictions with more than one currency, then you can create duplicate reinsurance programs for each currency. For example, an insurer has negotiated a set of reinsurance agreements for Europe in euros. The insurer 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 insurer creates one reinsurance program with the agreements in euros. The insurer creates another reinsurance program with the agreements in British pounds. The insurer 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 593
- *Configuration Guide*

Ceding premium in a multicurrency policy

The insurer 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 insurer 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 insurer cedes 10% of the premium to the reinsurer. If the currency is euros and the premium is 100€, the insurer cedes 10€ to the reinsurer. If the currency is U.S. dollars and the premium is \$100, the insurer 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, an insurer provides multicurrency policies that can have risks in different parts of the world. For risks in Europe, the insurer 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 in Reinsurance Management” on page 611

Multicurrency and basis units

Certain types of exposures, such as in General Liability, measure risk by the number of *basis units*. In some cases, insurers measure the basis units in currency: 100 units of *sales per \$1000*, for example. In other cases, the basis units are not measured in currency: 300 basis units of *area per acre*. In a General Liability submission, the **Policy Review** screen displays basis units for an exposure. Under **Exposure Value by Location**, each row has an **Exposure Type** and **Exposure Value** column. The **Exposure Type** displays the basis units for the exposure. The **Exposure Value** displays the number of basis units. This is just one example of a place where PolicyCenter displays basis units.

In a multicurrency system, the insurer does not necessarily match the currency of the basis unit with the policy or coverage currency. For example, one insurer’s actuarial tables for an exposure are only in U.S. dollars. The **Exposure Type** column displays **Sales per \$1000** regardless of the policy or coverage currency. Another insurer may change the currency of the basis units based on business rules.

In the base configuration, columns displaying basis unit are not part of the multicurrency product model configuration. You can configure the PolicyCenter user interface to display different basis unit strings in these columns by adding your business logic to the PCF file and associated code. The **Exposure Type** column strings are defined in the `class_code_basis.xml` system table. You may also need to configure the rating engine to rate the exposure in the correct currency.

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 PolicyPeriod Coverable Coverable delegates	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, <code>CommercialPropertyLine</code> entity provides an example.
PreferredSettlementCurrency	Account Contact PolicyPeriod	The preferred currency for settlement or billing.
CoverageCurrency	Cost Reinsurable Transaction	The coverage currency.
SettlementCurrency	Cost Transaction WorkersCompLine	The settlement currency.

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` configuration parameter setting is semi-permanent.

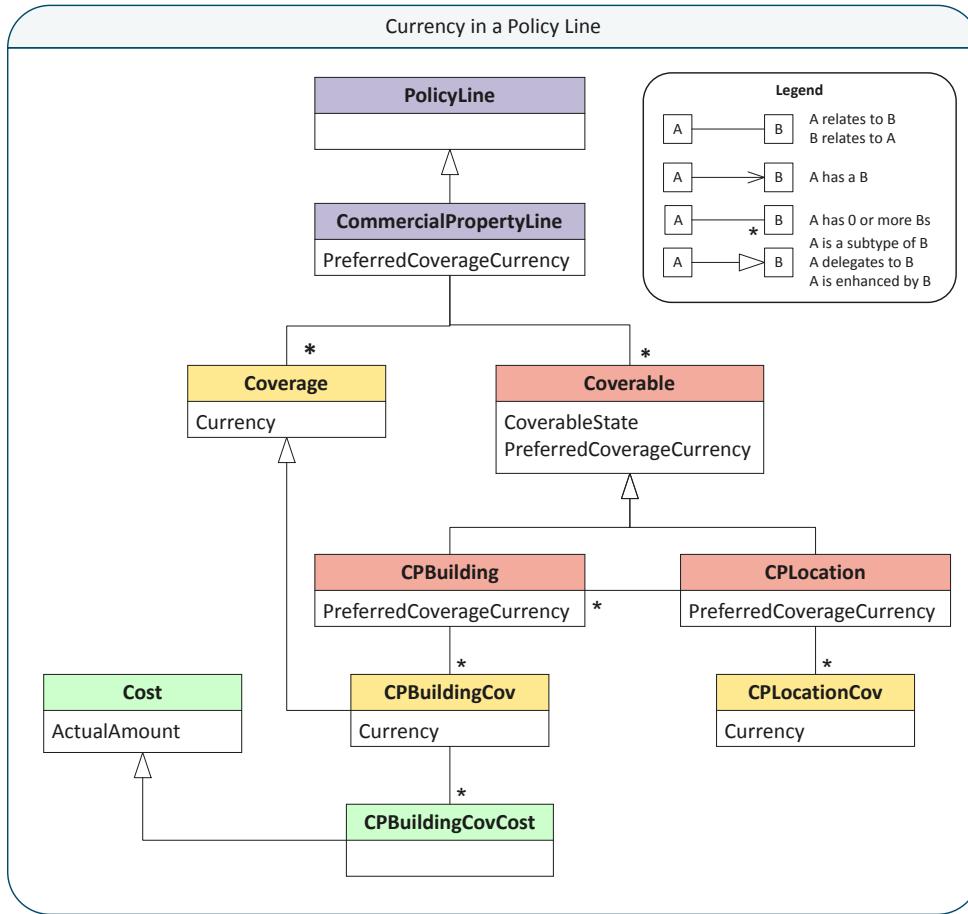
You can change the value of this parameter only once, from `SINGLE` to `MULTIPLE`. If you have changed the value and started the application, you must not later change it back. If you do change the value of `MultiCurrencyDisplayMode` back to `SINGLE`, subsequent attempts to start the server fail.

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 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 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 view and change the preferred settlement currency on the Account Summary screen.

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

- *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 Policy Info → Preferred Currency → Coverage field. The drop-down list contains the Available Coverage Currencies defined in Product Designer on the main page 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.

See also

- “Multicurrency and basis units” on page 492

Quote screen

The **Quote** screen displays all amounts in the settlement currency.

Continuing the example in “Multicurrency fields on policy transactions screens” on page 494, 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 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.

On the **Account→Billing** screen, the **View in BillingCenter** link opens the primary affiliated account in BillingCenter.

Multicurrency fields on the Organization screen

You can access producer organizations by navigating to the **Administration →Users & Security→Organizations** screen. On the **Agency Bill Plans** tab, you can choose among various agency bill plans from the billing system.

In a single currency system, you can specify one agency bill plan for each **Organization** in the **Agency Bill Plan** field on the **Basics** tab.

In a multicurrency system, each agency bill plan can offer one or more currencies. You can select more than one agency bill plan. For each agency bill plan, specify the currencies. For each producer organization, you can associate only one plan per currency. Therefore, if you select USD on Plan A, you cannot select USD on Plan B.

Multicurrency fields on the Producer Code screen

You can access producer codes by navigating to the **Administration →Users & Security→Producer Codes** screen. On the **Commission Plans** tab, you can choose among various commission plans from the billing system.

In a single currency system, you can specify one commission plan for each **Producer Code** in the **Commission Plan** field on the **Basics** tab.

In a multicurrency system, each commission plan can offer one or more currencies. You can select more than one commission plan. For each commission plan, specify the settlement currencies in which the producer can bind policies. For each producer code, you can associate only one plan per currency. Therefore, if you select USD in commission plan A, you cannot select USD in commission plan B.

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

- “Underwriting issues” on page 671
- “Multicurrency fields for underwriting authority” on page 497
- *Configuration Guide*

Risk Approval Details screen

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 screen

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.

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 and 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.

Archiving overview

Key features of archiving

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 processes identifies policy terms that need archiving, and then archives them.
- The archive batch process may skip or exclude some policy terms under the following conditions:
 - The policy terms are still associated with other active objects, such as open policy transactions, or jobs
 - The policy terms were recently retrieved from the archive
 - A user requested that a 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 **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.

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 archiving on your PolicyCenter configuration

Enabling archiving may significantly impact 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

More information on archiving

You can find more information about archiving in the following topics:

- *Upgrading archived or quote store entities using a version trigger* in the *Upgrade Guide*
- *Configuration Guide*
- *Installation Guide*
- *System Administration Guide*
- *Integration Guide*

Archiving policy terms

In PolicyCenter archiving, a complete `PolicyTerm` and all of its `PolicyPeriod` objects 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 archiving 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

- *Configuration Guide*

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 Desktop . 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 Archive Policy Term 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 true 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.

Run Archive Policy Term batch process

Before you begin

These steps assume that you have:

- Turned on archiving by setting the `ArchiveEnabled` parameter to true in `config.xml`.
- Not changed the settings of any other archive parameters in `config.xml`.
- Loaded the small sample data set. See the *Installation Guide*.
- Canceled the Ray Newton personal auto policy. See “Cancel a policy” on page 115.

About this task

This example is an exercise of archiving a policy in a PolicyCenter development environment. The exercise archives policy terms for the canceled Ray Newton policy.

Procedure

1. In PolicyCenter, 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. Select **Server Tools**→**Batch Process Info**.
4. Before the policy terms for the Ray Newton can be archived, 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.It is only necessary to run these batch processes manually while you are testing archiving by moving the system clock forward. In a production environment, these batch processes run on a regular schedule. In the default configuration, only policy transactions that have been canceled for a year or more are eligible for archiving. During a typical year, these batch processes run at least once.
5. Run the Archive Policy Term Batch process.

In this exercise, you run the Archive Policy Term batch process on demand. Typically in a production environment, the Archive Policy Term batch process runs on an infrequent schedule, such as weekly or monthly.
6. Select **Actions**→**Return to PolicyCenter**.
7. After the Archive Policy Term batch process completes, search for the Ray Newton auto policy to view the information that remains in PolicyCenter application database for an archived policy.

See also

- “Search archived policy periods” on page 503

Exclude policy from archiving

About this task

You can exclude an individual policy from archiving. If a policy is excluded from archiving, it will be excluded from the next run of the Archive Policy Term batch process.

Procedure

1. Navigate to a policy.
2. Select **Actions**→**Archiving**→**Enable/Disable**.
The **Enable/Disable Archiving** screen displays a message indicating whether or not archiving is enabled for this policy.
3. If the policy is enabled for archiving, select **Disable Archiving**, and optionally add a comment.

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.

Search archived policy periods

Before you begin

These steps assume that you have:

- Turned on archiving by setting the **ArchiveEnabled** parameter to true in **config.xml**.
- Not changed the settings of any other archive parameters in **config.xml**.
- Loaded the small sample data set. See the *Installation Guide*.
- Canceled the Ray Newton personal auto policy. See “Cancel a policy” on page 115.
- Advanced the system clock so that the Ray Newton policy is eligible for archiving
- Run the various archiving batch processes necessary to archive the canceled Ray Newton policy. See “Run Archive Policy Term batch process” on page 502.

About this task

This example is an exercise of searching for archived policies in a PolicyCenter development environment. The exercise searches for the Ray Newton policy, which has been archived.

Procedure

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.

See also

- “Retrieving archived policies” on page 504

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

- *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.

Steps to retrieve an archived policy

The example retrieves archived policy terms in a PolicyCenter development environment. Retrieval of archived policies occurs in two phases:

1. A user navigates to an archived policy and requests retrieval.
2. The Retrieve Policy Term batch process runs to retrieve archived policy terms for requested policies.

This example retrieves the policy terms for the archived Ray Newton policy.

Request retrieval of an archived policy

Before you begin

These steps assume that you have:

- Turned on archiving by setting the `ArchiveEnabled` parameter to true in `config.xml`.
- Not changed the settings of any other archive parameters in `config.xml`.
- Loaded the small sample data set. See the *Installation Guide*.
- Canceled the Ray Newton personal auto policy. See “Cancel a policy” on page 115.
- Advanced the system clock so that the Ray Newton policy is eligible for archiving
- Run the various types of archiving batch processing necessary to archive the canceled Ray Newton policy. See “Run Archive Policy Term batch process” on page 502.

Procedure

1. Navigate to an archived policy, such as the Ray Newton policy you archived in “Run Archive Policy Term batch process” on page 502.
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**.

Retrieve archived policies that have been requested

About this task

In these steps, you run the Retrieve Policy Term batch process on demand. Typically in a production environment, you run the Archive Policy Term batch process on a frequent schedule during the business day.

Procedure

1. Login as su.
2. Press Alt+Shift+T to access the Server Tools.
3. Navigate to the **Work Queue Info** screen.
4. In the row for **RestorePolicyTerm**, click **Run Writer**.
5. 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.

Personal data destruction

Note: The data destruction features described in these topics provide a set of features that help enable insurers to comply with some of their data destruction requirements. These requirements may be driven by insurers' policies and practices, as well as by their interpretation of various regulatory requirements. Such regulatory requirements may come from, for example, the European Union General Data Protection Regulation (GDPR) or the New York State Cybersecurity Requirements for Financial Services Companies law.

PolicyCenter supports destruction of some kinds of data. Destruction can mean either purging the data completely from the database or it can mean obfuscating data, making the original contents permanently unreadable.

Guidewire recognizes the need for insurers to be able to destroy personal information both on an on-demand basis or on a time-based basis. Destruction can be mandated by regulation or business practices, within the requirements of regulation, codes of conduct, or other business practices.

PolicyCenter provides support for data destruction and obfuscation that can be configured in Guidewire Studio.

See also

- *Configuration Guide*

Encapsulation of business logic for retention and destruction

Regulations, codes of conduct, and other generally accepted business practices vary from jurisdiction to jurisdiction. Additionally, business policies and interpretation of conflicting legal requirements vary from insurer to insurer. Therefore no single approach meets the needs of all insurers. To accommodate varying needs, PolicyCenter provides a configurable solution that captures business logic for retention and destruction in one place.

There is a configurable plugin that has access to the business objects to be removed through a root object. For example:

- Contact
- Policy
- Account

The examination of objects to be destroyed starts with the root object and traverses a graph of objects, enabling detailed examination of the business objects. You can mark requests requiring user review for those data destruction requests that require special handling, prior to the destruction actually occurring.

See also

- *Configuration Guide*

Notification of data protection officer on errors or conflicts

Requirements for destruction and for retention can conflict with each other. While the plugin class might be able to resolve conflicts in a generic way, situations can arise when the two sets of requirements are not reconcilable. Additionally, the data destruction process can encounter errors. In these situations, notification is done through a configurable plugin.

The default behavior of this plugin is that users with the new role of Data Protection Officer have an activity created describing the situation.

After the situation has been resolved, the destruction request can be queued again for reprocessing.

See also

- *Configuration Guide*

Wide-swath data destruction

In many situations, there is a need to destroy the personal data related to a specific business object. This data might be:

- A contact (a person)
- A record of a contract
- A record related to performance of a contract (policy)
- An aggregation of objects (such as a policy of an account)

These objects can affect many individual data objects. A single call allows the entirety of related data to be removed. In the case where these business objects are nested, a best-effort destruction is performed.

PolicyCenter components provide the ability to purge rows from the database for business objects such as `Contact`, `Policy`, and `Account`, and their related data. This approach is suitable for high-volume data destruction.

See also

- *Configuration Guide*

Individual-entity data destruction

While wide-swath data destruction meets the needs of the insurer in most cases, there are special cases where specific personal data cannot be deleted. For example, there might be database integrity concerns, or the data to be deleted, such as data for previous employee, might be related to a large number of policies.

In such cases, where individual instances of data cannot be deleted, PolicyCenter provides the ability to obfuscate data. Obfuscation can include wiping a field completely, replacing it with a neutral value, or replacing it with a unique, irreversible value.

The entities and fields to which obfuscation can be applied, as well as the method for determining the replacement value, are configurable.

See also

- *Configuration Guide*

Integration with other systems

PolicyCenter needs to be able to respond to data destruction requests from external systems, as well as have the ability to notify data consumers of data destruction.

PolicyCenter provides a web service that:

- Takes a reference to an individual contact.
- Takes application-specific action to destroy the data related to that contact.
- Reports back to the caller on the level of success of the request. Callers can query the status of a given request.

See also

- *Configuration Guide*

Notification of downstream systems

PolicyCenter provides a messaging system to assist you in ensuring that the destruction of personal data flows into systems connected with components. Additionally, you might need to notify outside organizations that process data on your behalf. The messaging system supports broadcasting personal data destruction response messages.

These messages are delivered by using the existing PolicyCenter guaranteed-delivery messaging system.

See also

- *Configuration Guide*

Part 6

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.

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. The rating execution is specifically implemented to calculate full term premium amounts and to prorate those amounts for the appropriate portion of the policy term.

The general rating process is implemented using multiple components.

Rating engine plugin is an interface definition and plugin implementation handles the premium calculation and proration process – the full rating execution. The base implementation uses a hierarchy of classes and flexible logic that product and line-specific implementations can use.

Costs are the containers for a monetary amount associated with a policy for premium, taxes, discounts and surcharges, and other amounts. In rating execution, costs are represented by the `CostData` Gosu object.

Rate flow is the logic in the rating engine plugin which orchestrates the premium calculation for a policy line. You can customize the rate flow implementation to meet your detailed requirements.

Rate books define a set of rate tables and rate routines. From a rate book, you can manage rate tables and rate routines, 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.

Rate tables provide definitions for a rate factor lookup table. Rate table content is defined within a rate book. 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 logic to calculate a premium amount or perform other operations in support of premium calculations. 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 contain parameters to pass to rate routines as discrete items or as part of a data structure. 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.

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 393
- *Integration Guide*

Key features of Rating Management

The key features of Guidewire Rating Management are:

- **Rate books** – Group related rate tables and rate routines into a rate book. Specify rate book properties including status and edition.
- **Editing** – Edit and manage rate books, rate tables, and rate routines in PolicyCenter.
- **Rate table import/export to spreadsheet** – Import and export rate table content to a spreadsheet in Microsoft Excel format. You can import existing rate table data into a draft rate book in PolicyCenter. You can export rate table data from PolicyCenter to a spreadsheet and make changes in a spreadsheet application. After making changes, import the changes back into PolicyCenter.
- **Rate book export to spreadsheet** – Export a rate book and its included rate tables and rate routines to a spreadsheet in Excel format. Import is not available.
- **Approval and migration** – Use tools for managing the lifecycle of rate changes, including capturing approvals, managing status, and migrating changes to test and production environments. For rate books, you can manage approval and migration with XML import and export.
- **Rate table versioning and selection** – Use availability criteria to specify which rate book applies for a given product, jurisdiction, effective date, underwriting company, and offering. Although 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** – Generates test policy periods so that you can see how changing the rate book and rate routines impacts policy premium. Choose the policies on which to see the impact by specifying 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 tables and rate routines from PolicyCenter to XML format. Use import and export to migrate rate books, rate tables, and rate routines between environments. See “Importing and exporting rate books to XML” on page 553.

- **Integration with the product model** – 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

In the base configuration, Rating Management provides rating for homeowners, personal auto, and commercial property lines of business. Through configuration, you can extend Rating Management to other lines of business.

See also

- *Configuration Guide*
- *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. 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

- *Installation Guide*

Rate flow design

Rate flow is the logic in the rating engine plugin which orchestrates the premium calculation for a policy line. You can customize the rate flow implementation to meet your detailed requirements, in particular, dependencies between calculations. You can modify the design of the rate flow in the base configuration in a number of ways.

Rate flow in the PCRatingPlugin

In the base implementation of Guidewire Rating Management, the `PCRatingPlugin` plugin implementation provides rating for Personal Auto and Commercial Property. These are the `PARatingEngine` and `CPRatingEngine` classes, respectively. Rate flows are a subset of the plugin and the `rateSlice` method is the container for the flow. After the `rateSlice` method completes, the `rateWindow` method calculates values that depend on all costs. In the base implementation, the example calculates taxes on the window. You must change this if you calculate taxes on the slice.

Alternate rate flow strategies

Some implementations require different rate flow strategies. For example, some implementations need to rate multi-line policies as a single component. To implement this, override the `rateOnly` method in the `AbstractRatingEngine` class to express the needed logic. The flow implementation will be similar to the `rateSlice` method but applied across multiple lines.

Rate book selection

The rate flow logic selects the rate book that will be used to access rate tables and rate routines. The general strategy for the flow is to select a rate book for usage, then access the objects on the policy to calculate full term amounts. In Personal Auto, a single rate book is matched to a policy. In Commercial Property, the rate book lookup happens for each coverable. The rate book is selected using the name, the reference date, and the rate book activation date. The

`CPRatingEngine` class selects a rate book for each coverable. Because the reference date is the date the coverable was added, a different rate book could apply in each case.

You can configure the rate flow implementation to select rate book based on different purposes. The example rating selects rate books by policy line. Another option is to put non-calculation logic (such as a rate routine that rates drivers) in a separate rate book from premium calculation.

Data dependencies

The rate flow orchestrates or orders the rating calculations to produce the premium amounts. If all premium calculations are independent of each other, then any ordering that fits the logical structure of the policy is valid. Where dependencies exist, you must modify the rate flow to reflect those requirements.

If you need two-pass rating, where the premium calculated in the first pass affects the final premium calculated in the second pass, you must modify the rate flow.

Rate tables in Rating Management

In Guidewire Rating Management, rate tables contain one or more rows with columns for parameter values. These columns are followed by the associated return factor value or values, if the rate table has multiple factors. Rate tables can optionally be associated with a particular policy line. You can also create generic rate tables that can apply to all policy lines.

In PolicyCenter, your rate table implementations may be significantly different from the legacy environments and exported format. If necessary, plan on implementing a conversion between the Rating Management import/export format and the format managed by business users.

Rate table overview

Parameter values can be of any data type. In the base configuration, values can be string, integer, decimal number, Boolean, and date. Factor values have the same data type. You can use decimal or integer values for numeric factor and rate. You can use strings for class definitions, tiers, and identifiers.

The following illustration shows a rate table with one factor.

Rate Table Content			
Export to Spreadsheet			
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	Kentucky	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.
3. Row – Provides parameter values and one or more rate factors.

Note: In the user interface for rate table content, rate factor columns are marked with an asterisk (*).

If the policy data matches parameter values in a row, PolicyCenter returns the factor value.

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 match, longest substring, less than or equal, greater than or equal, and interpolated match.

A parameter can be defined as accepting multiple values. Multiple values enables you to reduce the number of rows in rate table content. For example, a rate table definition has the same rate factor for a sedan or SUV in California, Washington, or Oregon. With multiple values, instead of six rows, you can represent this rate factor using one row.

You enable multiple values by selecting the **Allow Multiple Values** field in the rate table definition. You can select multiple values for a parameter if:

- You have selected a value provider.
- The matching rule is exact match.
- The data type is string.

Another parameter cannot depend upon a parameter that allows multiple values.

See also

- “Parameters tab of Rate Table Definition screen” on page 559
- “Parameters tab of Rate Table Definition screen” on page 559
- *Configuration Guide*

Value provider in rate table definition

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**. You can also specify a value provider on the **Factors** tab of a rate table definition.

The base configuration 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 rate factor in a rate table. The following rate table content shows a small example of the data.

Year minimum	Year maximum	Manufacturer	Model	Style	* Class code
2001	2004	Toyota	Prius		CCAH
		Toyota	Prius		IDAH
2011		Nissan	Leaf		JDAE
1993	1997	Volvo	850	Station Wagon	GWMR
1993	1997	Volvo	850	4-door Sedan	GDAR

Note: In the user interface for rate table content, rate factor columns are marked with an asterisk (*).

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 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 base configuration 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 suitable for in-memory lookup. See “Rate table lookup in memory or database” on page 519.
- 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 base configuration includes one sample custom physical table. To view this sample, open the CoverageFactor rate table definition. The custom physical table is specified in the **Physical Table** field in the **Basics** tab.

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 columns for:

- Coverage Code
- Coverage Term Code
- Coverage Term Option Code
- Jurisdiction
- Factor

For large rate tables, Guidewire recommends using a custom physical table to for efficient memory usage in the database.

If the rate table is to be used with database lookup, Guidewire recommends adding an index to your custom row to reduce retrieval time. For optimal performance, include the foreign key to the rate table, all the parameter values, and all the factor values in the index. PolicyCenter displays a warning if database lookup is used on a row which is not fully indexed.

See also

- The *Configuration Guide* for details on creating custom physical tables
- “Rate table lookup in memory or database” on page 519

Rate table lookup in memory or database

For each rate table, you can choose whether to load the rate table content into memory or access it from the database. By default, PolicyCenter loads rate table content into memory. Loading the rate table content in memory can provide quicker access.

Consider using database access only for very large tables which are used relatively rarely or if loading the table into memory degrades system performance or results in out-of-memory errors.

See also

- “Rate Book screen” on page 549
- *Configuration Guide*

Rate table normalization for overlapping ranges

In a rate table that contains a range parameter and any other parameter, the table may have ranges that overlap. For tables that fall within specified configuration parameters, PolicyCenter normalizes the rate table to speed up rate factor lookups. This rate table normalization removes the overlapping ranges by expanding the number of rows in the table. However, normalization uses more memory. Therefore, when you edit a normalized rate table, you can choose whether or not to normalize. Normalization does not change the rate table content that the user sees in **Rate Table Content** on the **Rate Table** screen. PolicyCenter creates the normalized table in memory only.

For example, a rate table has two parameters. The first parameter, age, is a range parameter. The second parameter is the jurisdiction. In a rate book, the rate table has the following values:

Row #	Minimum Age	Maximum Age	Jurisdiction	* Factor
1	1	5	California	1
2	3	6		2

Row #	Minimum Age	Maximum Age	Jurisdiction	* Factor
3	5	8	Oregon	3

Note: In the user interface for rate table content, rate factor columns are marked with an asterisk (*).

Ages from 3 to 5 potentially match rows 1 or 2. Ages from 5 to 6 potentially match rows 2 and 3. To remove the range overlaps, the table is normalized in memory as follows:

Row #	Minimum Age	Maximum Age	Jurisdiction	* Factor
1	1	3	California	1
2	3	5	California	1
3	3	5		2
4	5	6		2
5	5	6	Oregon	3
8	6	8	Oregon	3

Note: This is not a realistic example of a rate table. For example, a null match would result for a child older than 8. A child of age 2 in a jurisdiction not California would also generate a null match.

See also

- *Configuration Guide*

Rate table normalization user interface

If you view a normalized rate table in PolicyCenter, the following message appears above the rate table:

This table contains overlapping ranges which creates internally normalized rows...

If you edit the table, you can choose to not normalize the rate table by selecting **Reduce memory usage**.

Considerations for rate table size

Rate table size is an important factor to consider when designing rate tables. Rate tables have no inherent size limit within PolicyCenter, however very large tables (100,000+ rows, for example) present management issues during development and promotion. For large tables consider rate lookup, import/export, and business maintenance requirements.

To simplify lookup, you can split large tables into smaller tables, reducing the number of rows. For example, you can split a large table based on parameter domains. A vehicle table can be broken up by manufacturer or by year. You can create a rate routine function to select the correct table. When using functions, have the function do Rating Management lookup, not direct database lookup, to ensure that you access the correct version of the rate book.

Large rate tables with multiple factors can be normalized by splitting into two or more tables. See “Considerations for single or multiple factor rate tables” on page 523.

If you cannot split a very large table with 100,000+ rows, then consider creating a custom physical table. The **DefaultRateFactorRow** entity is very generic, but is inefficient with regards to storage, especially when exported to XML.

Another approach is the move the data out of a rate table and implement it as a system table. Design the system table with effective and expiration date columns. Similar to a rate book lookup, the lookup to system tables uses the reference dates. This approach requires custom configuration. Once you have the data in the external table, you can use utility functions in rate routines to lookup the tables.

Matching a factor in the rate table

After finding the most appropriate rate table, 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 in columns from left to right; relaxing parameters proceeds from right to left. 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 rightmost parameter from the matching criteria. The query matches a row in which the rightmost parameter is blank and all other parameters match. The relaxing of 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

Note: In the user interface for rate table content, rate factor columns are marked with an asterisk (*).

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 in a rate table

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 from left to right in ascending priority order. Therefore **Jurisdiction** appears to the left of **Amount**.

In PolicyCenter, the rate table has **Jurisdiction** and **Amount** parameter columns and **Factor** rate factor column. The following table has additional **Row** and **Description** columns.

Row #	Jurisdic- tion (exact)	Amount (>=)	* Factor	Description
1		1.00		Matches any salary if both Amount and Jurisdiction 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 Amount is specified, all parameters to the left must be specified.
3	CA		1.25	If Amount is relaxed, matches any salary in the California jurisdiction.
4	CA	30,000	1.50	Matches a salary in the amount of \$30,000 to less than \$50,000 in the California jurisdiction. The policy value \$30,000 is greater than or equal to the parameter value.
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 Amount is relaxed.

Row #	Jurisdiction	Amount	* Factor	Description
		(\geq)		
		(exact)		
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 (CA) 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 (FL) 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.

See also

- “Parameters tab of Rate Table Definition screen” on page 559

Example of finding a factor with a range parameter in a rate table

This example is similar to “Example of finding a factor in a rate table” on page 521. However, this example 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 (\geq)** and **Max (\leq)**. The **Min (\geq)** argument is priority 10, the **Max (\leq)** argument is priority 20. PolicyCenter relaxes the range parameter as a single unit.

In PolicyCenter, the rate table has **Min (\geq)**, **Max (\leq)**, and *** Factor** columns. The following table has additional **Row** and **Description** columns.

Row #	Jurisdiction	Min (\geq)	Max (\leq)	* Factor	Description
		(exact)			
1			1.00		Matches any salary when both the salary range parameter and Jurisdiction 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 (\geq)** and **Max (\leq)** 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 (\geq)** and **Max (\leq)**, is relaxed. A person in Florida who has a \$35,000 matches row 1 after both salary and **Jurisdiction** are relaxed.

Performance considerations

The data in the previous table can also be represented using a less than or equal match (\leq). However, the range match is more efficient than the \leq match. When looking up [CA, 15,000] with range parameters, row 3 is the only row that matches the range. With the \leq parameter, the lookup matches rows 3 through 6, eventually returning row 3 as the best match.

Row #	Jurisdiction	Salary (<=)	* Factor (exact)
1			1.00
2	NY		1.25
3	CA	30,000	1.25
4	CA	50,000	1.50
5	CA	80,000	1.75
6	CA		2.00

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

Note: In the user interface for rate table content, rate factor columns are marked with an asterisk (*).

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.

Considerations for single or multiple factor rate tables

Rate table definitions can have one or more factor columns. Consider using multiple factors only when all factors represent the same data. For example, in the sample data, the AutomaticRenewalCap rate table in the Personal Auto line has two factors: Amount and Percent. Different business logic (in rate routine steps) is associated with each factor. However, both factors represent the same data, the renewal cap, presented in one factor as an amount and as a percentage in the other.

If factors represent different data, consider modeling that data as a parameter. For example, you have a spreadsheet with one parameter and seven factors for different data: one for each Canadian province. The following table provides an example:

Tier	* Alberta	* Ontario	* Quebec	* ...
Silver	0.8	1.0	1.2	...
Gold	1.0	1.2	1.4	...

Tier	* Alberta	* Ontario	* Quebec	* ...
...

In the example above, a rate routine that retrieves the factor for all provinces requires an IF statement with seven conditions, one for each province. Consider modelling the province as a parameter instead of as a factor. However, if this approach works better for you, then you can simplify the rate routine with a function. The function takes the province as a parameter and does the rate table lookup to get the correct return factor. If rate table size is an issue, consider creating separate rate tables. For example, each province has its own rate table.

When the province is modelled as a parameter instead of as a factor, the rate routine is simpler because you only have to pass in the lookup province. The following table provides an example:

Tier	Province	* Factor
Silver	Alberta	0.8
Silver	Ontario	1.0
Silver	Quebec	1.2
Gold	Alberta	1.0
Gold	Ontario	1.2
Gold	Quebec	1.4
...

The rate routine requires only one statement to access the factor. You improve the readability of the rate routine but increase the number of table rows.

Database normalization can be helpful in designing rate tables. Consider normalization if you denormalized a rate table because the legacy system could not process the normalized tables fast enough. For example, your vehicle risk factor is equal to the year factor multiplied by the model factor. Your previous implementation used a single rate table with rows for each year and vehicle combination. If there are five years and five models, the rate table contains 25 rows. Using normalization, you can implement this using two five row rate tables. One table contains the risk factor for the year and another table contains the risk factors for the model.

Note: This database normalization is different than rate table normalization for overlapping ranges.

See “Rate table normalization for overlapping ranges” on page 519.

Rate table with interpolated rate factor

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. In addition, the interpolated parameter can be defined as with or without relaxing. To learn how this affects matching, see “Rate table with interpolated rate factor” on page 524.

The data type of the interpolated parameter must be a numeric type: either decimal or integer. The data type of the interpolated 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	* Interpolation Factor
1				14

Row #	Jurisdiction	Interpolated parameter	Discount code	* Interpolation Factor
2	New Jersey			13
3	New Jersey	0		10
4	New Jersey	6		8
5	New Jersey	10		6
6	New Jersey	10	Silver	4

Note: In the user interface for rate table content, rate factor columns are marked with an asterisk (*).

Jurisdiction and Discount Code are exact match parameters.

The only way you can match row 6 is with the exact inputs: Jurisdiction is New Jersey, Interpolated Parameter is 10, and Discount Code is Silver. Hereafter, these values are represented as {New Jersey, 10, Silver}.

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 you have {New Jersey, 6, Silver}. By relaxing the Discount Code parameter, the input values match row 4.

If the policy values are {New Jersey, 8, Silver}, then Discount Code is relaxed. Rows 4 and 5 are the bounds for interpolation because New Jersey matches, and the policy value 8 is between 6 and 10.

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:

```
IF = (
    ( (PV - IP1) ÷ (IP2 - IP1) )
    X (F2 - F1)
)
+ 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 {New Jersey, 8, –}. The interpolated factor is computed by using rows 4 and 5 in the rate table. The formula is:

```
IF = ( ( (8 - 6) ÷ (10 - 6) ) X (6 - 8) ) + 8
```

Since the formula is a bit complicated, the following lines show in detail how to compute the input factor:

```
IF = ( ( 2 ÷ 4 ) X (6 - 8) ) + 8
IF = (      %      X ( -2 ) ) + 8
IF = (          -1      ) + 8
IF = 7
```

Interpolated parameters with and without relaxing

Interpolated parameters have an additional type of relaxing. When you define an interpolated parameter, you specify it as either with or without relaxing. An interpolated parameter without relaxing requires that both bounds for the input parameter be found at the same level of relaxation. An interpolated parameter with relaxing allows the bounds to be found at different levels of relaxation. With relaxing, the algorithm finds the bound above or below the input parameter at some level of relaxation, then further relaxes the query to find the other bound.

To see how relaxing affects the bounds, the following rate table has an interpolated parameter and an exact match Discount Code. The policy values of the input parameters are {5, Silver}.

Row #	Interpolated parameter	Discount code	* Interpolation Factor
1	0		14
2	10	Silver	12
3	10	Gold	10
4	20	Silver	8
5	20	Gold	6
6	100		4

Without relaxing – If you define the interpolated parameter without relaxing, rows 1 and 6 are the bounds. These rows match at the same relaxation level (by relaxing the Discount Code). Although Silver matches the Discount Code in rows 2 and 4, the Interpolated Parameter 5 is not between 10 and 20. Therefore, these rows are not the bounds. When Discount Code is relaxed, rows 1 and 6 are the bounds. Rows 1 and 6 are the bounds because they do not specify a Discount Code, and 5 falls between the Interpolated Parameter values 0 and 100.

With relaxing – If you define the interpolated parameter with relaxing, rows 1 and 2 are the bounds. On the first pass to find a match, row 2 is the first bound because the Discount Code matches and the Interpolated Parameter value of 10 is greater than 5. To find the second bound, the Discount Code is relaxed, and row 1 provides the second bound.

Rate routines in Rating Management

In Guidewire Rating Management, use rate routines to implement rating algorithms that calculate properties on the cost for coverages, taxes, and other policy costs. 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 535
- *Configuration Guide*

Rate routine design

For each insurance company, *rate filings* describe in detail how premium is calculated. When filings are required, filings can be the basis for rate routines. The rate filing documentation often does not provide all the information necessary for designing the rate routine. A subject matter expert can transform the rate filing into requirements for a rate routine.

When designing rate routines from rate filings, follow these guidelines:

- Limit the rate routine to calculations that will be versioned within a rate book.
- Keep the rate routine focussed on things that the business user needs to see in the rating worksheet. Treat the rate routine as a document of the calculation rather than a general programming language. When the rate routine executes, the output document is the rating worksheet. Design the rate routine so that the rating worksheet reveals the premium calculation.

For example, there is no need to put a polynomial calculation, having multiple parameters, in a rate routine.

Instead, use a utility function and pass in the parameters. Business users do not need to know the detailed steps in the polynomial calculation. They just need to know that the polynomial calculation is done as a step in rate routine and passed certain parameters.

- The primary output of rating is the cost of a single coverage. Many secondary outputs are also possible as described in “Rate routines that do not calculate properties on the cost” on page 527.

Rate routines that do not calculate properties on the cost

Many 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 properties on the cost. These 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. You can use this rate routine as a model for creating your own rate routines that do not set 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 the `rateSlice` method in `PARatingEngine.gs`.

A no-cost rate routine uses a parameter set that does not include cost. In no-cost parameter sets, **Include Cost** is not selected. The **PA Assign Driver** rate routine uses the **PA Driver Assignment Parameter Set** parameter set.

See also

- “Add parameter set in Rating Management” on page 580
- *Configuration Guide*

Using rate routines for non-premium requirements

You can use rate routines that do not calculate properties on the cost for calculations that are secondary to calculating premium. Some examples of these calculations are:

- Retain the discount/surcharge amounts applied to the premium calculation. These amounts are retained in variables that are passed into subsequent rate routines.
- Calculate premium for coverage term values other than the selected one so that the customer can see differences in premium.
- Implement rating requirements, such as two-pass rating, within the rate routine.

Use two-pass rating if you have to run the rating algorithms twice on the policy data with small differences. For example, in some types of capping, you rate the new policy using the old rate book and then the new rate book. Or, you have to find the highest risk driver by running the rate routine on each driver to get the highest risk. With that risk, you can then process the premium calculation.

Although you can implement these non-premium requirements in the rate routine, you risk making the rate routine more complex than the calculation requires. Consider moving the non-premium calculations to a separate rate

routine. Assess the trade-off between complexity and requirements on a case by case basis. An example is driver assignment.

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 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	--		Base Rate and Adjusted Rate Calculation
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	vehicleType-Factor	← table:Vehicle Type Factor	
5	vehicleCost-Factor	← 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.

#	Instruction	Op (Operand)) Line comment
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	--	Term Amount Determination	
13	IF	AdjustedRate < table:Min Premium AND Vehicle.VehicleType = VehicleType.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 566

Considerations for rate routine functions

Functions provide a mechanism to execute arbitrary Gosu code from within a rate routine. Use functions:

- To externalize logic that cannot be expressed in a rate routine. For example, looping through array elements.
- For calculations that are secondary to the core purpose of calculating a rate. For example, calculating complex rate factors.
- To simplify rate routines, and to retain the rate routine clarity as documentation of the rate calculation. Use worksheet logging to expose function actions to the rating worksheet when appropriate.

If the business user needs to see the steps in the calculation or the algorithm needs to be versioned, then consider using a rate routine.

Functions do not have the same data access limitations as the rate routine editor does so they enable access all policy data. You can use functions to perform complicated rate table lookups, giving you greater control of the argument to parameter association. Functions can dynamically determine the rate table that you want to lookup.

Rate routine functions provide logic where the business user does not need to examine the interim steps of the calculation. For example, you can implement a function to calculate the average age of drivers on all vehicles in a policy. In the rating worksheet, the business user does not need to see how the function calculates the average age. In addition, computing the average age is always the same and does not need to be versioned over time.

See also

- “Functions in rate routine steps” on page 575

Accessing entity properties and class extensions

Rate routines can access entity properties, including virtual properties, and class extensions. You can create and use entity properties and extensions to provide rate routine access to that data without exposing the business logic that derives it. An example of this in the base configuration is the `Person.Age` virtual property. The rate routine can access this property, and calculating the age is handled by the entity.

Parameter sets in Rating Management

Parameter sets contain one or more parameters which are accessible to Guidewire Rating Management. A parameter set can be specific to a policy line or available to all policy lines. The parameter set usually includes a policy line parameter which provides access other parts of the policy. You can also specify parameters for entities such as coverages or rating entities, such as rating information and rate date.

Rate routines

In a rate routine step, you can use parameters to access policy information, such whether the car has antilock brakes. Each rate routine specifies a parameter set. The rate routine can access the parameters for its calculations.

Rate table definition

In a rate table definition, you must specify a parameter set which contains the default sources of arguments for the rate table *lookup parameters*. The lookup parameters correspond to lookup columns in the rate table. On the **Parameters** tab, you define lookup parameters. On the **Argument Sources** tab you specify a default argument source for each lookup parameter. You specify values for the argument source by accessing parameters in the parameter set. When you specify a rate table in a rate routine step, the default values of the lookup parameters are the default argument sources. You can change the argument source for each lookup parameter. If the rate table definition has multiple argument source sets, you can change the argument source set for this rate table lookup.

See also

- “Working with parameter sets in Rating Management” on page 579
- *Configuration Guide*

Parameter set design

Parameter set definition and maintenance has an impact on multiple components of the rating solution. Consider this impact as you determine the parameter configuration.

Parameter sets provide the data interface between the rate flow and individual rate routines. In the base configuration, example parameter sets show three types of content.

Policy objects

Parameters can be policy objects such as policy line, vehicle, or coverage. Use policy objects to identify the object being rated when the policy may contain multiple objects of this type. The rate flow iterates over all vehicles and calls a routine per vehicle. The rate flow creates the parameter set and during the first iteration the vehicle parameter represents the first vehicle. During the following iteration, the vehicle parameter represents the second vehicle. Keep in mind that generic objects, like `Coverage`, do not provide access to specific values like limit amounts on a particular type of coverage.

Atomic values

Parameters can be atomic values such as `PreviousTermAmount`. These types of parameters are relative easy to define and use. However, atomic parameters may become difficult to manage if you define too many. Consider using data objects instead.

Data objects

Parameters can be data objects such as `DriverAssignmentInfo` through which you can access properties on the object. Data objects can be arbitrarily complex while keeping the parameter set simple. However, the parameter set list does not indicate what information is being passed in at first glance. The user has to look at the drop-down list to see the object's properties.

Combine similar parameter sets with wrappers

You can use *wrappers* to access parameters in a line of business that are typed differently. For example, coverage has a generic version and is typed specifically for each coverage. With a wrapper for coverage, passing in a coverage to a rate routine enables access the coverage terms off the coverage. The wrapper calls code that selects a coverage based upon characteristic of the policy.

See also

- “Combine similar rate routines and parameter sets” on page 545

Example with wrappers

In the base configuration, the **CP Coverage Parameter Set With Wrapped Coverages** parameter set uses a wrapper. The **Coverage** parameter uses the `CPCoverageWrapper` wrapper to select either the **Business Personal Property Coverage** or the **Building Coverage**. In Product Designer, you can view these coverages in the **Commercial Property Line**.

The **CP Coverage Premium Algorithm** rate routine uses the **CP Coverage Parameter Set With Wrapped Coverages** parameter set. In this rate routine, the basis is assigned the value of the coverage limit. The wrapper calculates the value of the limit based upon whether the coverage is **Business Personal Property Coverage** or **Building Coverage**.

Rate books in Rating Management

In Guidewire Rating Management, 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. A rate book is uniquely identified by code, edition, and policy line. As a unit, the rate book can be versioned and promoted to active status for use in production.

You cannot modify the rate routines included in active rate books. PolicyCenter limits the types of changes that you can make to rate tables used in active rate books. PolicyCenter will not let you make changes that affect rating in an active rate book.

In rate books, you can define qualifying attributes which appear as the **Policy Criteria** on the rate book screen:

- Policy line – Required. Specify a policy line or applies to all policy lines.
- Jurisdiction – Optional. Specify a jurisdiction or applies to all jurisdictions.
- Underwriting company – Optional. Specify an underwriting company or applies to all underwriting companies.
- Offering – Optional. Specify an offering or applies to all offerings.

A rate book has the following availability attributes:

- Policy effective or coverage reference date – Required.
- Activation date – The date that the rate book was activated into production.

Managing rate books and rate tables

Rate books contain a group of related rate tables. More precisely, each rate book edition groups a specific set of rate table versions. When you edit rate table content in a rate book, PolicyCenter creates a new version of that rate table. The rate book includes the new rate table. You can manage rate book editions in PolicyCenter.

As soon as you activate a rate book by moving it to active status, the rate book is available for production use. To ensure the integrity and auditability of policies rated with the active rate book, you cannot make changes to an active rate book and its contents.

If any rate table in an active rate book requires a change to the content, you must create a new edition of the rate book. In the new edition, you can update the rate table. 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.

Sometimes changes to the rate book are necessary. Changes to a rate book can be driven by regulatory changes or business reasons. Some changes add rows to one or more rate tables. Other changes add or remove columns (parameters) from rate tables. Certain changes require that you make a new edition of the rate book. In addition, the rate book status affects the types of changes you can make.

Changes to rate routines

You can change a rate routine if it is not included in a rate book or it is only included in draft rate books. You cannot change a rate routine that is used in a promoted rate book.

Considerations for rate book maintainability

Rate books contain a set of rate tables and rate routines.

The size of the rate book, gauged by the number and complexity of included rate tables and rate routines, impacts maintainability and regression testing. Smaller rate books may support a quicker development cycle, more effective unit testing, and the ability to constrain necessary regression testing to affected areas of the rating process.

The simplest choice is to use a single rate book for each policy line and share logic through the inclusion of common rate routines and rate table. For example, you have common rate routines and rate tables. Each policy line has a rate book which includes these common elements and also includes rate routines and rate tables specific to the line.

However, some implementations may benefit from a higher level decomposition strategy. For example you can have dedicated rate books which include only include rate routines and rate tables that apply to all lines. By implementing cascaded lookup, you can have country-wide rate tables and rate routines in one country-wide rate book and the state deviations in state-specific rate books.

You can also separate independent streams of logic into separate rate books. For example, within a single policy line, individual rate books contain the rate routines and rate tables for calculating specific types of costs. Or within a multi-line product, each policy line has its own rate book.

Rate book storage self-contained

A *self-contained* rate book owns all of the included rate tables.

You can make a rate book self-contained if both of the following are true:

- The rate book is in Stage status
- The rate book is referencing rate tables

IMPORTANT Making the rate book self-contained changes the relationship between the rate book and included rate tables. When a rate book references a rate table, that rate table is stored in another rate book which owns it. When making the rate book self-contained, PolicyCenter replaces the reference with a copy of the rate table. Therefore, the storage is duplicated in both rate books. You cannot undo this change.

Business examples

In a development environment, you have created multiple editions of a rate book which you do not wish to replicate in production. You are now up to edition 10, and the rate book has been tested on the staging server and is ready for production. Edition 10 contains links back to various previous editions of the rate book. You only wish to deploy the 10th edition onto the production server. Therefore, on the staging server, you make the rate book self-contained.

In a production environment, you have a rate book with 5 editions. Due to regulatory changes, you must implement major changes in rate tables in the 5th edition of the rate book. Because the changes are far-reaching, you no longer wish to reference rate tables. When you move the rate table back to stage status, you make the rate book self-contained.

Changes to parameters in rate table definitions

Within certain restrictions, you can add and delete parameters from rate table definitions without creating a new edition of the rate book. This flexibility enables you to change the rate table definition without adversely affecting existing usages of the rate table. Whether or not the rate table definition is *in-use* affects the types of changes you can make.

Note: A rate table definition is *in-use* when it is included in a rate book. When a rate table is included in a rate book, PolicyCenter instantiates the rate table by populating the rate table content.

Because you cannot make changes to an active rate book and its contents, PolicyCenter prevents changes to rate table definitions that affect the rating of an active rate book. For example, you can add new higher priority parameters to a rate table definition used by an active rate book. However, you cannot edit rate table content in an active rate book. The rate table content will always be null for the new parameter's column.

See also

- “Adding parameters to an in-use rate table definition” on page 587

Adding parameters to rate table definitions

You can add parameters to an existing rate table definition. Several factors, including whether the rate table definition is *in-use* or not, affect how and where you can add parameters. For example, in some circumstances, you can only add parameters of higher priority. Or to add parameters, you may have to define a new argument source set.

Note: A *not-in-use* rate table is not included in any rate book. A not-in-use rate table can be included in a rate routine. That rate routine can even be included in a rate book. The rate table is not-in-use as long as it is not included in a rate book.

The following table describes when you can add parameters to a rate table definition. The table includes corner cases where an not-in-use rate table is referenced in a rate routine.

Rate table definition is...	Can add parameter?
Not in use and...	<ul style="list-style-type: none">Not referenced by any rate routine. Yes, you can add a parameter. PolicyCenter automatically adds the parameter to all existing argument source sets.
	<ul style="list-style-type: none">Referenced only in rate routine, and rate routine not included in any rate book.Referenced in rate routine, and rate Yes, you can add a parameter. PolicyCenter automatically adds the parameter to all existing argument source sets.

Rate table definition is...	Can add parameter?
routine is only included in draft rate books.	You must go into each rate routine and refresh the rate table lookup. When you view the rate routine, PolicyCenter displays a message that it is unable to reconcile the definition of the rate table. To resolve this issue, edit the rate routine, and select the rate table again. PolicyCenter displays a message that new parameters have been added. Click OK to accept the change.
• Referenced in rate routine included in a promoted rate book	<p>Yes, you can add a parameter.</p> <p>To use the new parameter, you must create a new argument source set. PolicyCenter displays a warning message when you add the parameter.</p> <p>PolicyCenter automatically adds the parameter to all new argument source sets.</p> <p>PolicyCenter also adds the parameter to existing argument source sets if the argument source sets are not referenced in rate routines included in promoted rate books.</p>
In-use and...	<p>Used only draft rate books</p> <p>Yes, you can add a parameter.</p> <p>PolicyCenter automatically adds the parameter to all existing argument source sets.</p> <p>You must go into each rate routine and refresh the rate table lookup. When you view the rate routine, PolicyCenter displays a message that it is unable to reconcile the definition of the rate table. To resolve this issue, edit the rate routine, and select the rate table again. PolicyCenter displays a message that new parameters have been added. Click OK to accept the change.</p>
Used in promoted rate book	<p>Yes, but you can only add a parameter with higher priority than existing parameters.</p> <p>To use the new parameter, you must create a new argument source set. PolicyCenter displays a warning message when you add the parameter.</p>

Deleting parameter from rate table definitions

You cannot delete a parameter from a rate table definition if any of the following are true:

- The rate table definition is in-use and the parameter has a column value in the rate table.
- The parameter is referenced in a rate routine included in a promoted rate book.
- The rate table definition is in-use in a promoted rate book or referenced by a rate routine in a promoted rate book.

See also

- “Adding parameters to an in-use rate table definition” on page 587

Editing parameter details in a rate table definition

You can edit parameter details except if any of the following apply:

- The rate table definition is in-use and the parameter has a column value in the rate table.
- The parameter is referenced in a rate routine included in a promoted rate book.
- The rate table definition is in-use in a promoted rate book or referenced by a rate routine in a promoted rate book.
- The parameter is used in a rate table look up in any rate routine.

Changing factors in rate table definitions

Your ability to add, edit, or delete rate factors in rate table definitions is similar to parameters with these exceptions:

- Factors cannot be added, deleted, or edited if the rate table definition is in-use in a promoted rate book
- Factors cannot be referenced by a rate routine in a promoted rate book

See also

- “Changes to parameters in rate table definitions” on page 533

Other types of changes in rate table definitions

You cannot edit the code, physical table, or policy line of a rate table definition if either of the following is true:

- The rate table definition is in-use.
- The rate table definition is referenced in a rate table lookup in a rate routine.

These fields appear on the **Basics** tab of the rate table definition. You can edit the description.

Versioning in Rating Management

Consider the full scope of versioning early, particularly for rate books, rate tables, rate routines, and other components. Rating Management provides versioning of rating components as described below.

Rate flow

Logic defined in Gosu code in the rate flow has no inherent versioning other than source control strategies. Avoid versioning in the rate flow. Versioning in the rate flow must be done with conditional logic, which can become difficult to support.

Rate tables

Versioned by their inclusion in a particulate edition of a rate book. Rate books are selected by the rate flow implementation based on date criteria from the policy data and the values set in the rate book content.

Rate routines

Versioned outside the rate book. The rate book includes a specific version of the rate routine.

Rate routine functions

Functions are not versioned in rate books and the example implementations do not include versioning. Suggested approaches to handle this issue are:

- **No versioning** – Some functions, such as functions that do not affect premium calculations, do not require versioning. For example, a function for logging.
- **Versioning through method parameter** – In the function definition, include a parameter for version number. The rate routine passes in the version number when it calls the function. For example, `mySampleFunction(1.0, Vehicle)`. It is up to your implementation of the function to run the appropriate code for that version.
- **Versioning by changing the name of the function** – Create a new version of the function name by creating a new function with a version string or qualifier appended to the name. For example, name the next version of `getYoungestDriver` function `getYoungestDriver_v1`. Or name a version for motorcycles `getYoungestDriverForMotorcycles`.
- **Versioning the implementation class** – This is the most complex approach. Replace the mechanism that creates the function container class with an implementation that includes awareness of the version required, possibly using the rate book edition as a parameter. This approach requires modification of Gosu classes that implement the rating engine.

Selecting the rate book edition during policy rating

In Guidewire Rating Management, a rating query selects the most appropriate rate book, and rate book edition, to use for policy rating. The rate book contains the rate tables and rate routines used in policy rating.

In addition, if a rate book has *cascaded lookup* enabled, the query uses a cascaded, or hierarchical list of, rate books. If the most appropriate rate book does not contain a specific rate table or rate routine, then the query looks in the cascaded rate books.

The rating query executes the query in these stages:

1. Determine and retrieve the most appropriate rate book based on the coverage and policy attributes and the “Rate book matching process” on page 536. If cascaded lookup is enabled, then also append the list of cascaded rate books.
2. When evaluating the rate routine, retrieve the correct factor from the rate table, based on the input values provided.

Rate book matching process

Rating Management looks for rate routines and rate tables in:

- **The most appropriate rate book only** if cascaded lookup is disabled in the most appropriate rate book.
- **Cascaded rate books** if cascaded lookup is enabled in the most appropriate rate book.

The rate book matching process differs between selecting the most appropriate rate book and selecting the cascaded rate books.

In either case, however, the query uses certain required attributes when selecting candidate rate books. The following required attributes must match for a rate book edition to be a candidate:

- Effective/Expiration Dates – The coverage **Reference Date** must be later than or equal to the rate book’s effective date (**On or After**) and earlier than the expiration date (**Before**). For more information about the coverage reference date, see 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 the *Configuration Guide*.

The candidate rate books are further filtered to find matches. Different filtering algorithms are used for selecting the most appropriate rate book and cascaded rate books.

When selecting rate books, rate book matching uses the following algorithm:

1. The query retrieves a list of candidate rate books that match required attributes, as described above.
2. The query further filters down the list based on the values of the optional attributes: policy line, jurisdiction, underwriting company, and offering. The query uses one filtering algorithm to find the most appropriate rate book. The query uses another filtering algorithm to find the cascaded rate books.
3. If a single matching rate book is found, the query returns it.
4. If multiple matching rate books are found, the query returns the rate book with the latest activation date that is earlier than the rate as of date. For example, the query can find multiple matches when there is more than one edition of a rate book.

Filtering for the most appropriate rate book

When selecting the most appropriate rate book, if an exact match is not found, *relax* the optional attributes by looking for `null` in the optional attributes. The relaxing rules build an ordered list of rate books by relaxing the optional attributes. A value of `null` matches all policy lines, or other optional attribute.

PolicyCenter relaxes the optional attributes by first relaxing offering, then underwriting company, jurisdiction, and finally policy line. The following table shows the order in which PolicyCenter relaxes the optional attributes:

Exact match			
Policy line	Jurisdiction	Underwriting company	Offering
Relax offering			
Policy line	Jurisdiction	Underwriting company	<code>null</code>
Relax underwriting company			
Policy line	Jurisdiction	<code>null</code>	Offering

Policy line	Jurisdiction	null	null
Relax jurisdiction			
Policy line	null	Underwriting company	Offering
Policy line	null	Underwriting company	null
Policy line	null	null	Offering
Policy line	null	null	null
Relax policy line			
null	Jurisdiction	Underwriting company	Offering
null	Jurisdiction	Underwriting company	null
null	Jurisdiction	null	Offering
null	Jurisdiction	null	null
Relax policy line and jurisdiction			
null	null	Underwriting company	Offering
null	null	Underwriting company	null
null	null	null	Offering
Relax policy line, jurisdiction, underwriting company, and offering			
null	null	null	null

You can change the rate book matching logic through configuration.

See also

- *Configuration Guide*

Filtering for cascaded lookup rate books

If cascaded lookup is enabled in the most appropriate rate book, then rate book matching can find rate tables and rate routines in cascaded rate books. If the most appropriate rate book does not contain a matching rate table or rate routine then rate book matching looks in the cascaded rate books.

With the group label, you restrict cascaded lookup to only match rate books in the same group as the rate book originating the cascaded lookup. A rate book with no group matches only rate books with no group. The group label only applies to cascaded lookup. On the Rate Book screen, you specify **Group** in **Rate Book Details**.

Rate book cascading continues until it finds a match or reaches a rate book with cascaded lookup disabled.

The list of cascaded rate book is selected by relaxing the optional attributes by looking for null in the optional attributes. PolicyCenter relaxes the optional attributes by first relaxing offering, then underwriting company, jurisdiction, and finally policy line. The following table shows the order in which PolicyCenter relaxes the optional attributes for cascaded lookup:

Policy line	Jurisdiction	Underwriting Company	Offering
Policy line	Jurisdiction	Underwriting Company	null
Policy line	Jurisdiction	null	null
Policy line	null	Underwriting Company	null
Policy line	null	null	null
null	null	null	null

The `matchersForHierarchy` method of `RatingQueryFacade.gs` defines the relaxation hierarchy for cascaded rate books.

Example of cascaded lookup in rate book

With cascaded lookup, the rate query uses relaxation to find rate books containing a matching rate routine or rate table. By default, new rate books have cascaded lookup enabled.

This example shows cascaded lookup in personal auto and assumes the rate books have no group and have cascaded lookup enabled. For simplicity, the example omits the required fields and underwriting company and offering.

Assume that PolicyCenter has the following rate books:

Rate book	Rate routine	Rate table	Policy line	Jurisdiction
Country-wide Rate Book	Vehicle Coverage Premium Algorithm	Base Rate		
California Rate Book		Base Rate	Personal Auto	California
Massachusetts Rate Book	Vehicle Coverage Premium Algorithm		Personal Auto	Massachusetts

Assume that all instances of the Vehicle Coverage Premium Algorithm rate routine access the Base Rate rate table.

To rate a vehicle coverage on a California policy, the rating query finds the most appropriate rate book: California Rate Book. The California rate book does not contain a Vehicle Coverage Premium Algorithm rate routine.

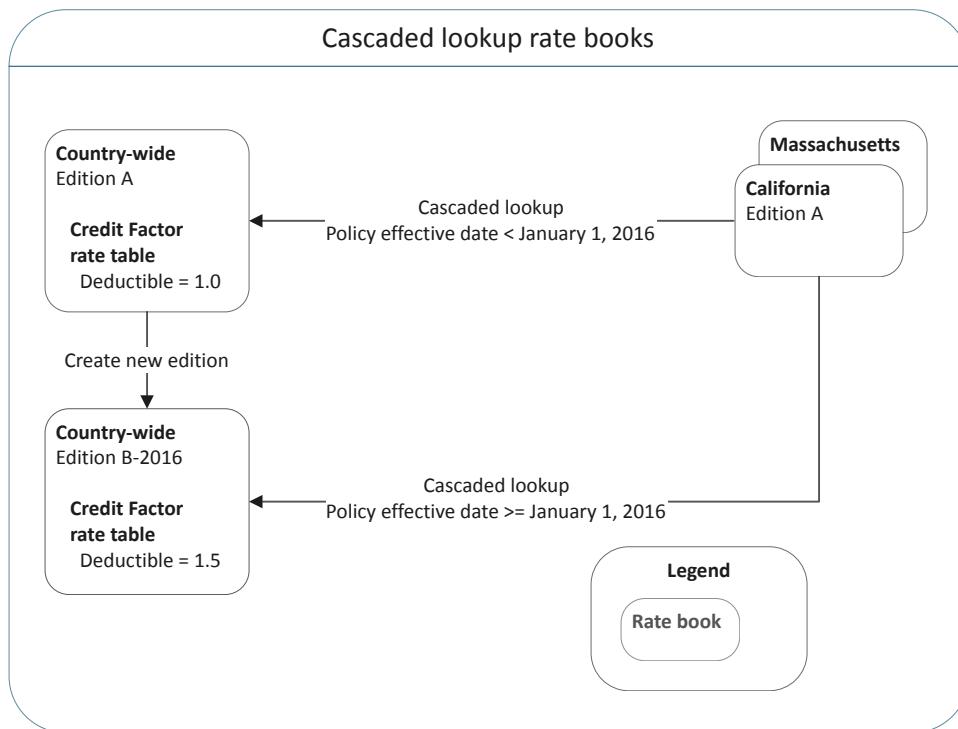
Therefore, the rating query uses cascaded lookup to find the rate routine. The query relaxes jurisdiction then policy line to find the Vehicle Coverage Premium Algorithm rate routine in the Country-wide Rate Book. The rate routine accesses the Base Rate rate table. The query finds the Base Rate rate table in the California Rate Book.

To rate a vehicle coverage on a Massachusetts policy, the rating query finds the most appropriate rate book: Massachusetts Rate Book. The Massachusetts Rate Book contains a Vehicle Coverage Premium Algorithm rate routine. This rate routine accesses the Base Rate rate table, but the Massachusetts Rate Book does not contain the Base Rate rate table. Therefore, the query uses cascaded lookup and relaxes jurisdiction then policy line to find the Base Rate rate table in the Country-wide Rate Book.

Cascaded lookup and rate book edition

Cascaded lookup enables state-specific rate books to access the latest country-wide content. Without cascading, the state-specific editions must contain the full data from country-wide rate book along with state-specific data. Without cascaded lookup, a change to country-wide data must be replicated in each state-specific edition.

For example, the current rate book edition is A. For 2016, you must implement changes to the credit factor rate table. This rate table is in the country-wide rate book. To implement these changes, create a new B-2016 edition of the country-wide rate book effective on January 1, 2016. In the B-2016 edition, update the rate table with the 2016 factors. Now, a California or Massachusetts policy with an effective date on or after January 1, 2016 uses the credit factor rate table in the country-wide B-2016 edition. California and Massachusetts policies effective prior to January 1, 2016 use the credit factor rate table in the country-wide A edition.



Overlapping effective policy periods and rate-as-of date in rate book

Rate books can have overlapping effective policy periods.

How can overlapping effective policy periods 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 policy periods, why does the rate book selection process not automatically select the replacement or most recently activated rate book? In many cases, regulations require that an insurer 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 1/1/2020 and activated 10/15/2019 so that it is available for 1/1/2020 renewals which begin processing 11/1/2019.
- On 11/15/2019, Policy A effective 1/15/2020 is rated and bound. Rate Book v1 is used, and the rate-as-of date is 11/15/2019.
- On 12/1/2019, Rate Book v2 with updated 2020 rates is introduced. Rate Book v2 is also effective 1/1/2020 but activated 12/1/2019.
- On 12/15/2019, Policy B effective 1/15/2020 is rated and bound. Rate Book v2 is used, and the rate-as-of date is 12/15/2019.
- If both Policy A and Policy B were endorsed effective 2/1/2020, 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 = 11/15/2019) picks up Rate Book v1 (activation date = 10/15/2019) and NOT Rate Book v2 (activation date = 12/1/2019).
 - Policy B (rate-as-of date = 12/15/2019) picks up Rate Book v2 (activation date = 12/1/2019).

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.

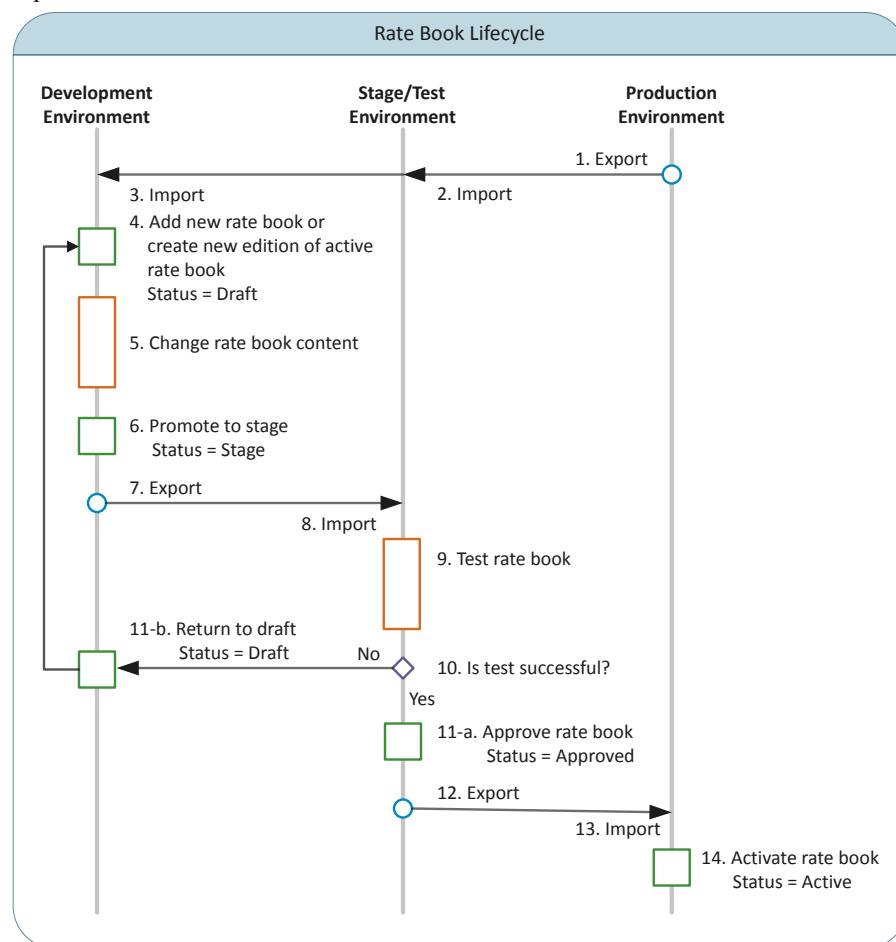
In the example, you can force Policy A to be re-rated with Rate Book v2. Simply initiate a policy change with the same effective date as the policy and change the rate-as-of date to later than 12/1/2019.

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 insurers 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 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 with Rating Management

In the development environment that includes Guidewire Rating Management, 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 the *System Administration Guide*.

Stage environment with Rating Management

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 insurers 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 the *Configuration Guide*.

Production environment with Rating Management

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 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 the *Configuration Guide* for details of the included roles and permissions. You can configure this to meet your specific needs.

Rating worksheets in Rating Management

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:

Instruction	Result	Op (Operand) Operand value
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

Instruction	Result	Op (Operand) Operand value
9	x	table:CovFactor	(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

- “View rating worksheets” on page 579
- *Configuration Guide*

Extracting and purging rating worksheets

Every time a user generates a rating worksheet, a copy of that rating worksheet gets stored in the PolicyCenter database. To improve PolicyCenter performance, PolicyCenter provides processes for extracting rating worksheet data and removing rating worksheets from the database. This is not an end user feature and is only accessible through **Server Tools**.

The Extract Rating Worksheets batch process extracts the rating worksheet data to files and marks worksheets for purging.

The Purge Rating Worksheets batch process removes worksheet container objects that are marked for purging and older than a specified number of days (90 days in the base configuration). In the base configuration, you must enable this batch process.

See also

- *Configuration Guide*

Testing Rating Management

Rating Management includes impact testing which enables you to assess the impact that changes to the rate book and rate routines have on policy premium for a group of policies. This topic also provides some guidelines to improve the testability of your Rating Management implementation.

Impact testing for Rating Management

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 base 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 base 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 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

See also

- “Working with impact testing in Rating Management” on page 581
- *Configuration Guide*

General guidelines for testability of Rating Management

This topic provides some guidelines to improve the testability of your Rating Management implementation.

Exclude EffDated objects from functions

Automated unit testing of code can be complicated if testing requires a policy. Creating a policy presents significant overhead to test creation and execution. Therefore, avoid passing effective dated objects to business logic where possible. When a function operates on atomic values, then unit tests can be small and simple.

Include data capture mechanisms

Test result or test execution data provides a useful way verify quality. PolicyCenter and the Guidewire platform provides data analysis tools and logging, but you can add additional logic to measure and verify the implementation.

Internal profiling

The Guidewire Profiler provides detailed information regarding which components consume time while processing a request. In addition, you can configure custom entry points. This data is particularly useful for profiling database interactions.

Performance measurement

If the base implementation does not support your performance data capture needs, you can configure data capture code at appropriate points. Results can be written to a log file, stored in the application database, or both. Since this approach is limited in scope, it can be used to capture and analyze production behavior without excess overhead.

Verifying inputs and outputs

PolicyCenter provides several ways to capture data inputs and outputs for test verification.

- Impact testing enables you to capture large amounts of policy and rating information for analysis.
- Rating worksheets display rating algorithm calculations on a policy. Generating worksheets may result in slower rating performance. Consider adding a way to turn worksheets on or off based on user input or specific policy conditions.

Through configuration, you can also develop other data capture mechanisms. Identify these needs early their implementation so that they can be built into the design.

Reducing Rating Management components

Rate books, rate table definitions, rate routines, and parameter sets are *Rating Management components*. There are several ways to reduce Rating Management components in your implementation.

Rating Management component applies to all

You can define Rating Management components with policy line, underwriting company, jurisdiction, or offering that applies to all by specifying <**applies to all**>. For example, you can define a rate routine that applies to all underwriting companies.

For each rating management component, the following table shows which properties can apply to all. The table also lists examples in the sample data that apply to all.

Rating Management component	Properties that can apply to all...	Example in sample data
Rate book	<ul style="list-style-type: none">• Policy Line• Underwriting Company• Jurisdiction• Offering	Generic RTM Demo Rating Code: gx_rtm_demo_rating
Rate table definition	<ul style="list-style-type: none">• Policy Line	GenericStateTax Code: gx_state_tax
Rate routine	<ul style="list-style-type: none">• Policy Line• Jurisdiction	Generic State Tax Calculation Code: gx_state_tax_rr
Parameter set	<ul style="list-style-type: none">• Policy Line	State Tax Parameter Set Code: GenericStateTaxParamSet

Policy line applies to all

Rating Management components can apply to a specific policy line, such as the Personal Auto line of business. Alternatively, Rating Management components can apply to all policy lines, when you set the **Policy Line** to <**applies to all**>. In code, if a component applies to all policy lines, the policy line is `null`.

Line-specific Rating Management components can reference Rating Management components that apply to all policy lines as well as components defined for the same policy line. For example, a rate book with Personal Auto as its policy line can reference rate tables that apply to all policy lines and Personal Auto rate tables. The Personal Auto rate book cannot reference Commercial Property rate tables.

Rating Management components that apply to all policy lines can only reference components that apply to all policy lines. They cannot reference Rating Management components defined for a specific policy line. For example, a rate book that applies to all policy lines can only reference rate tables that apply to all policy lines. The rate book that applies to all policy lines cannot reference Commercial Property rate tables.

Combine similar rate routines and parameter sets

Parameter set *wrappers* enable you to combine similar parameter sets and rate routines.

You can use wrappers for accessing parameters in a line of business that typed differently. For example, a single parameter set can use a wrapper to select the coverage by some characteristic of the policy. Wrappers are implemented in Gosu code.

In addition, rate routines that are essentially the same can be combined into a single rate routine by using a parameter set with wrappers.

The sample data has examples using wrappers including:

Rating object	Name	Code
Parameter set	CP Coverage Parameter Set With Wrapped Coverages	CPCoverageWrapperParamSet
Rate routine	CP Coverage Premium Algorithm	cp_cov_premium_rr

See also

- “Combine similar parameter sets with wrappers” on page 531

Rating Management user interface

This topic describes how to work with Guidewire Rating Management in the PolicyCenter user interface.

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 the *Integration Guide*. In the base configuration, PolicyCenter has the permissions that enable you to work with Guidewire Rating Management. For more information, see the *Configuration Guide*.

Working with rate books

You can search for, view, edit, copy, add, delete, import or export rate books.

Search for rate book

Procedure

1. Select **Administration**→**Rating**→**Rate Books**.

PolicyCenter displays the **Rate Books Search** screen.

2. Specify any of the following search parameters:

- **Policy Line** – Select a policy line. Default is **<not specified>**, which matches rate books regardless of which policy line the rate book applies to. For example, **<not specified>** matches rate books in Personal Auto Line, Commercial Property Line, and all other policy lines. Select **<applies to all>** to match rate books that apply to all policy lines. See “Rating Management component applies to all” on page 545.
- **Code** – Matches rate book codes that contain the search criteria.
- **Name** – Matches rate book names that contain the search criteria.
- **Jurisdiction** – **<not specified>** matches rate books regardless of jurisdiction. Select **<applies to all>** to match rate books that apply to all jurisdictions.
- **Underwriting Company** – **<not specified>** matches rate books regardless of underwriting company. Select **<applies to all>** to match rate books that apply to all underwriting companies.
- **Offering** – **<not specified>** matches rate books regardless of offering. Select **<applies to all>** matches rate books that apply to all offerings.

- **Status – <not specified>** matches rate books regardless of status.
- **Policy Effective or Coverage Reference 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**.

The **Activity** column for the rate book displays **Export To Spreadsheet** while the rate book is being exported to spreadsheet.

4. Click the hypertext link in the **Name** column to view or edit any of the returned rate books in the Rate Book screen.

When viewing the Rate Book screen, **Policy Criteria** set to **<applies to all>** appear blank.

Add new rate book

Before you begin

You must have the `ratebookview` and `ratebookedit` permissions.

Procedure

1. Add a new rate book using one of two methods.
 - Click **New Rate Book** in the **Search Results** pane.
 - View an existing rate book and click **Create New Edition**.

PolicyCenter displays the Rate Book screen in edit mode. If you created this rate book by clicking **Create New Edition**, the rate book edition and the effective (**On or After**) and expiration (**Before**) dates are blank. All other fields, including rate tables and rate routines, 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 screen” on page 549.

2. If this is a new rate book, select the **Policy Line**.
3. Select **<applies to all>** if the rate book can apply to all policy lines.
4. Select **<applies to all>** for all **Policy Criteria**.

Delete rate book

Before you begin

You must have the `ratebookview` and `ratebookedit` permissions to delete a rate book.

About this task

You can delete a rate book if it is in **Draft** status.

Procedure

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**. The rate book is deleted, and you are done.
Otherwise, click the link in the **Name** 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.

Rate Book screen

Use the Rate Book screen to view and work with rate books. You can only update rate books in **Draft** status. See “Rate book lifecycle and moving to production” on page 540 for more information about rate book statuses.

You must have the `ratebookview` permission to view this screen and the `ratebookedit` permission to make changes.

The Rate Book screen is made up of the following sections: the **Rate Book Details**, the **Included Rate Tables**, and the **Included Rate Routines**.

Rate book details

The Rate Book screen contains the following fields:

Field	Description
Name	A human-readable name for the rate book. You cannot edit the name of an existing rate book. When viewing a rate book, the name appears at the top of the screen and this field is hidden. This field is localizable. Required.
Rate Book Details	
Code	A code for the rate book. The combination of code and edition must be unique within the specified policy line. Required.
Edition	A string to represent the edition for the rate book. The format of the edition is specific to each insurer and may contain characters and numbers, such as 1.0, 2011, ISO Circular 4138:2010. Required.
Description	A description for the rate book. This field is localizable.
Status	Set to Draft for new rate books. Set to other statuses based on rate book actions. See “Rate book status and available actions” on page 551.
Cascaded Look-up	Use cascaded lookup to find matching rate tables and rate routines in a hierarchical list of rate books. If you specify a group, cascaded lookup finds the most appropriate rate book among rate books that specify the same group. If you do not specify a group, then cascaded lookup finds the most appropriate rate book in rate books with no group.
Group	Specify a group for cascaded lookup.
Last Status Change Date	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 “Selecting the rate book edition during policy rating” on page 535 for more details on the use of activation date in the rating queries.
Activation Date	
Last Updated By	The user who last updated the rate book.
Last Updated Time	The time of the last update.
Storage	Looks at the Usage column. Values are: <ul style="list-style-type: none">• Self-contained – The rate book owns all rate tables.• Contains References – This rate book references one or more rate tables. Referenced rate tables appear with Referencing in the Usage column on the Included Rate Tables table of the Rate Book screen. If the rate book is in Stage status,  appears after this label. Click this button to make the rate table self-contained. This change cannot be undone. See “Rate book storage self-contained” on page 532.
Policy Criteria	
Policy Line	A drop-down list of policy lines defined in the product model. Select <applies to all> if the rate book applies to all policy lines. See “Rating Management component applies to all” on page 545. Required.
Underwriting Company	A drop-down list of underwriting companies defined in PolicyCenter. Select <applies to all> if the rate book applies to all underwriting companies.

Jurisdiction	A drop-down list of jurisdictions defined in PolicyCenter. Select <applies to all> if the rate book applies to all jurisdictions.
Offering	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. Select <applies to all> if the rate book applies to all offerings.
Policy Effective or Coverage Reference Date	<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"> • The effective date of the policy term • The date that the coverable object was added to the policy within the current policy term • The date that the coverage was added to the policy within the current policy term <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"> • On or After – The effective date of the policy must be equal to or greater than this date. Required. • Before – The effective date of the policy must be less than this date. <p>For more information about the reference date of the coverage, see the <i>Product Model Guide</i>.</p>
Renewal Effective Date	<p>Specify the range that the policy renewal effective date must fall within:</p> <ul style="list-style-type: none"> • On or After – The policy, coverable, or coverage reference date must be equal to or greater than this date. Required. • Before – 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 Before in Policy Effective or Coverage Reference Date.

When viewing the Rate Book screen, **Policy Criteria** set to <applies to all> appear blank.

Including 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 Rate Tables** panel 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**.

In the **Lookup** column, you can specify whether the rate table is loaded into memory (default), or is accessed from the database. Use this setting to improve rate table lookup performance.

In some cases the **Lookup** value cannot be changed. The value cannot be changed if:

- The rate book is not in **Draft** status. Rating management consults the lookup setting when promoting the rate book status from **Draft** to **Stage**. Therefore, you can only change this parameter when the rate book is in **Draft** status.
- The rate table is referenced from another rate book. The **Lookup** value appears in parentheses, for example, **(Memory)**. A rate table can have only one setting for lookup. You can only change the lookup setting in the rate book that includes, rather than references (**Referencing in Usage**), the rate table. The rate book must be in **Draft** status.

See also

- “Rate table lookup in memory or database” on page 519

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, select the version from the drop-down list, 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.</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: Edit, Delete, Promote to Stage (if the rate book includes at least one rate table), and Export to Spreadsheet.</p> <p>You cannot export a rate book to XML that is in this status.</p>
Stage	<p>Data entry is complete and the rate book is ready for testing.</p> <p>Available actions: Approve Rate Book, Return to Draft, Export to Spreadsheet, and Export to XML.</p>
Approved	<p>Testing is complete and the rate book is approved and ready to be moved to production.</p> <p>Available actions: Return to Draft, Activate Rate Book, Export to Spreadsheet, and Export to XML.</p>
Active	<p>The rate book is available for production use.</p> <p>Available actions: Create New Edition, Export to Spreadsheet, Export to XML.</p>

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	<p>View the Administration→Rating menu items.</p> <p>On the rate book Search Results, click the rate book code hypertext link to view the rate book.</p> <p>You must have the ratebookview permission.</p>
New Rate Book	<p>In Administration→Rating→Rate Books, click New Rate Book in Search Result to add a new rate book.</p> <p>You must have the ratebookedit permission.</p>
Rate Book Details	
Edit	<p>On the Rate Book screen, click Edit to edit the current rate book.</p> <p>The rate book must be in Draft status and you must have the ratebookedit permission.</p>
Delete	<p>On the Rate Book screen, click Delete to delete the current rate book.</p> <p>The rate book must be in Draft status and you must have the ratebookedit permission.</p>
Promote to Stage	<p>On the Rate Book screen, click Promote to Stage to change the status from Draft to Stage.</p> <p>The rate book must be in Draft status, must have at least one included rate table, and you must have the ratebookedit permission.</p> <p>Before changing the status, the system verifies that none of the included rate tables are empty.</p>
Approve Rate Book	<p>On the Rate Book screen, click Approve Rate Book to change the status from Stage to Approved.</p> <p>The rate book must be in Stage status and you must have the ratebookapprove permission.</p>
Return to Draft	<p>On the Rate Book screen, click Return to Draft to change the status back to Draft to make changes to the rate book or included rate tables and rate routines.</p> <p>The rate book must be in Stage or Approved status and the user must have the ratebookedit permission.</p>

Action	Description
Activate Rate Book	On the Rate Book screen, click Activate Rate Book to change the status from Approved to Active . The rate book must be in Approved status and user must have ratebookapprove permission. The system asks for a confirmation of this action before proceeding.
Create New Edition	On the Rate Book screen, click Create New Edition to create a new edition of the current rate book. This action: <ul style="list-style-type: none"> • Creates a new rate book. • Copies all of the rate book fields except version and effective and expiration dates. • Creates referencing relationships to all included rate table versions. See “Managing rate books and rate tables” on page 531 for details on referencing relationships and versions. The rate book must be in Active status and you must have the ratebookedit permission.

See also

- *Configuration Guide*

Merge rate books

About this task

You can merge two rate books with the same policy line. The merged rate book contains rate book properties, rate routines, and rate tables from each rate book. If a property has a different value in each rate book, then you can choose which value to use. For differences in rate table content or rate routines, you can select which rate table content or rate routine to use. If a rate table exists in one rate book only, then the rate table is copied to the new rate book. The same applies for rate routines. In the merged rate book, the rate table usage is referencing.

To merge two rate books:

Procedure

1. Navigate to the **Rate Books** screen.
2. Select two rate books with the same policy line, and click **Merge**.
You can only select rate books on the same page of **Search Results**.
PolicyCenter displays the **Merge Rate Books: Resolve Conflicts** screen.
3. Enter an **Edition**.

The edition is a string of characters and number that must be unique for the code. The initial code of the merged rate book is the same as the first rate book. If the rate books have different codes, you can select which code to use.

The **Result** column displays icons showing how properties, rate tables, and rate routines differ in the two rate books.

Icon	Description
✓	This rate book property, rate routine, or rate table is the same in both rate books.
◊	Merge conflicts. Use the radio buttons to select values for the merged rate book.
⊖	This rate routine or rate table exists only in the first rate book.
⊕	This rate routine or rate table exists only in the second rate book.
⊖⊕	Merge conflicts. This rate table in the first rate book has been modified from the common parent. This rate table in the second rate book is the same as the common parent.

Icon	Description
	Merge conflicts. This rate table in the second rate book has been modified from the common parent. This rate table in the first rate book is the same as the common parent.

4. Review the results. Using the radio buttons, select a value for each merge conflict.

5. Click **Complete Merge** to merge the two rate books.

PolicyCenter merges the rate books and displays the **Edit Rate Book** screen. You can make changes to the merged rate book, including changing the code, name, and edition.

Export rate book to spreadsheet

About this task

You can export a rate book and its included rate tables and rate routines to a spreadsheet in Excel format. The rate book can be in any status. The first tab displays rate book details, policy criteria, and links to included rate tables and rate routines. Each rate table and rate routine appears in its own tab. PolicyCenter always exports dates in rate books to YYYY-MM-DD format.

To export a rate book to spreadsheet format:

Procedure

1. Navigate to a rate book.
2. Click **Export→Export to Spreadsheet** in the Rate Book screen.

While the rate book is exporting:

- PolicyCenter displays **Export to Spreadsheet** and a status bar indicating the progress of the export. You can navigate to other screens without affecting the export progress.
- On the **Rate Books** screen, the **Activity** column for the rate book displays **Export to spreadsheet in progress** while the spreadsheet is exporting.
- You cannot edit, delete, or change the status of the rate book.

After the export completes, on the Rate Book screen **Last Spreadsheet Exported** displays the time of the last export.

After export, the **Export** drop-down menu is not available. To export again, navigate away from and back to the Rate Book screen.

3. Click **Download→Download Spreadsheet** to download the spreadsheet to your local computer.

See also

- “Edit rate table content in Excel” on page 564
- *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.

On import PolicyCenter checks that the imported rate book does not conflict with data already defined in the target environment. The XML can contain additional parameters in a rate table definition. However, the XML cannot contain fewer parameters. For example this can occur when you need to update a rate table definition in multiple instances. In one instance of PolicyCenter, you update the rate table definition by adding a new parameter, then export the containing rate book to XML. You then import that XML to second PolicyCenter instance.

If a rate book contains rate tables that are referencing, not owned, then the exported XML contains links to the rate table content, rather than the rate table content.

If you are moving to a system which does not contain the referenced rate table content, you can change the rate table storage to be self-contained. When you export a self-contained rate book, the XML includes the rate table content.

IMPORTANT Making the rate book self-contained changes the relationship between the rate book and included rate tables. When a rate book references a rate table, that rate table is stored in another rate book which owns it. When making the rate book self-contained, PolicyCenter replaces the reference with a copy of the rate table. Therefore, the storage is duplicated in both rate books. You cannot undo this change.

See also

- “Rate book storage self-contained” on page 532
- *Configuration Guide*

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. Developers can work on their changes separately then merge their changes back to PolicyCenter. See “Merge rate books” on page 552.
- Make certain that each environment has a unique public ID prefix. See 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.
- Do not use export and import rate books to move rate books between versions of PolicyCenter. The data format may change between versions. Therefore, use the upgrade procedure which upgrades rate books to the new PolicyCenter version. Then you can export rate books from the newly upgraded version to another PolicyCenter instance with the same version.

Export rate book to XML

Before you begin

You must have `ratebookview` permission. The rate book can be in any status except **Draft**.

About this task

These steps export the current rate book to XML so that you can move it to another environment.

The rate book is exported to XML. The XML includes the rate routines and the rate table data for all owned rate tables. The XML includes links to referenced rate tables.

Procedure

1. Navigate to the Rate Book screen.
2. (Optional) In the **Storage** field, click  to make the rate book self-contained. In a self-contained rate book, all rate table storage is owned rather than referenced.

IMPORTANT

You cannot undo this action. Therefore, before making the rate book self-contained, be aware of the implications. See “Rate book storage self-contained” on page 532.

This button appears if both of the following are true:

- The rate book is in Stage status
- **Storage** displays **Contains References**

3. Click **Export→Export to XML**.

While the rate book is exporting:

- PolicyCenter displays **Export to XML** and a status bar indicating the progress of the export. You can navigate to other screens without affecting the export progress.
- On the **Rate Books** screen, the **Activity** column for the rate book displays **Export to XML in progress** during the export.
- You cannot edit, delete, or change the status of the rate book.

After the export completes, on the Rate Book screen **Last XML Exported** displays the time of the last export.

After export, the **Export** drop-down menu is not available. To export again, navigate away from and back to the Rate Book screen.

4. Click **Download→Download XML** to download the XML to your local computer.

Import rate book from XML

About this task

You can import a rate book previously exported to XML format. 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 `MaximumFileSize` parameter.

Procedure

1. Navigate to the **Search Results** pane of the **Rate Books** screen.
2. Click **Import from XML** to import an exported rate book.
3. View import warnings or errors by clicking a link on the screen. These warnings and errors also appear in the log file.

See also

- “Delete rate book” on page 548
- *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, edition, 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.

Importing rate tables from spreadsheet

You can export a rate book to spreadsheet format, make changes, and import the rate tables back into PolicyCenter. The rate book is exported to spreadsheet in Excel format.

To export rate tables in a rate book, export the rate book as described in “Export rate book to spreadsheet” on page 553. The exported content includes the rate book details, included rate tables, and included rate routines. You can make changes to the rate table content.

While editing the rate book in PolicyCenter, you can import one or more rate tables from spreadsheet. If the spreadsheet contains errors, then the import fails. You can export errors back to a spreadsheet. Rate tables with errors have a red tab, and rows with errors are highlighted in yellow.

Edit rate table in spreadsheet format

Procedure

1. In PolicyCenter, export a rate book, such CP RTM Demo Rating, to spreadsheet, by following the instructions in “Export rate book to spreadsheet” on page 553.
In the Included Rate Tables tab, the **Lookup** value appears in parentheses if the rate table is referenced by the rate book. Otherwise the rate table is owned by the rate book. Note the name of a referenced rate table such as **BaseRate** with the **Lookup** value (**Memory**).
2. In the spreadsheet, edit the data in a referenced rate table. For example, in the **BaseRate** rate table on the **cp_coverage_base_rate** tab, change the **BaseRate** of a value.
3. Save your changes to the spreadsheet.

Import rate tables

Procedure

1. In PolicyCenter, navigate to a rate book, such as CP RTM Demo Rating.
2. Click **Edit**.
3. Select **Import** in the **Included Rate Tables** tab.
4. On the **Import Table from Spreadsheet** popup, browse to select the spreadsheet you exported and edited.
5. Click **OK**.
6. In the **Included Rate Tables** tab, notice that the **Lookup** column on the **BaseRate** rate table. Because you edited this rate table, the value is now **Memory** without parentheses, indicating that the rate book will own this rate table. The **Pending Import** column indicates whether a table will be added or modified. A message above **Rate Book Details** displays how many table will be added or imported.
7. To proceed with the import, click **Update**.

If an included rate table contains a different number of factors in PolicyCenter than in the spreadsheet, import fails. To fix this problem, add or remove factors from the spreadsheet to match the rate table in PolicyCenter.

Next steps

If an import error occurs, continue with “Continue import of rate table with error” on page 557.

Import large rate table from spreadsheet

About this task

When importing a large rate table, Guidewire recommends that you import the rate table on the Rate Table screen. Importing on the **New Rate Book** screen may result in normalizing range errors.

Procedure

1. In PolicyCenter, create a new rate book on the **Rate Books** screen.
2. In **New Rate Book** screen under **Available Rate Tables**, select the rate table and add it to the rate book.
3. Click **Update**.
4. Click the rate table link in **Included Rate Tables**. PolicyCenter displays the Rate Table screen.
5. Under **Rate Table Content**, click **Import** to import the rate table data.

Continue import of rate table with error

Before you begin

Perform these steps if, as described in “Import rate tables” on page 556, import rate tables resulted in an error.

Procedure

1. In the spreadsheet, add an error. For example, in the BaseRate table, first Coverage Code, change the Cause of Loss Code and Cause of Loss Display Name from Basic to Special.
2. Save your changes.
3. In PolicyCenter, import the spreadsheet you just edited.
PolicyCenter displays a message indicating errors which prevented import.
4. Click **Export Errors**.
PolicyCenter exports the rate tables back to a spreadsheet.
5. In the spreadsheet, navigate to the red tab.
6. Fix lines with errors highlighted in yellow.

Working with rate table definitions

You can search for, view, edit, copy, add, or delete rate table definitions.

Search for rate table definition

Procedure

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 <not specified>, which matches rate tables regardless of which policy line the rate table applies to. Select <applies to all> to match rate tables that can apply to all policy lines.
 - **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 **Name** and then by **Code**. **Policy Line** set to <applies to all> appears blank in the search results.
4. Click the hypertext link in the **Name** column to view or edit any of the returned rate table definitions in the **Rate Table Definition** editor.

Add a new rate table definition

Procedure

Add a new rate table definition 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 edit rate table definitions.

Basics tab of rate table definition screen

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. Select <applies to all> if the rate table definition can apply to all policy lines.
Physical Table	The physical rate table/entity to map this definition to. Can use the generic physical table or reference a custom physical table. To use the generic physical table, enter DefaultRateFactorRow . See “Physical tables and entities for rate table definitions” on page 518 for details on generic and custom physical tables. In the base configuration, the DefaultRateFactorRow entity represents the default physical table. The base configuration also include custom physical table represented by the CoverageRateFactorRow 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 or used in a rate routine, you can update only the **Name** and **Description** fields.

Parameters tab of Rate Table Definition screen

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.

For more information on making changes to parameters, see “Changes to parameters in rate table definitions” on page 533.

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 Match 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 Match 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.
- **Greater Than Or Equal Match** – The policy value must be greater than or equal to the parameter value. If multiple matches exist, the closest parameter value matches. For example, a rate table has rows for values 1 through 10. The policy value 5.5 is greater than or equal to rows 1 through 5. Therefore, row 5 matches.
- **Greater Than Match** – The policy value must be greater than the parameter value.
- **Less Than Or Equal Match** – The policy value must be less than or equal to the parameter value. If the policy value matches multiple parameter values, the closest parameter value matches. For example, a rate table has rows for parameter values 1 through 10. The policy value 5.5 is less than or equal to rows 6 through 10. Therefore, row 6 matches.
- **Less Than Match** – The policy value must be less than the parameter value.
- **Interpolation - with Relax** – If the policy value matches the parameter value exactly, return the factor. If the policy value falls between two parameters values, then the interpolated factor is proportionally between the two factors. Interpolation with relaxing allows the bounds to be found at different levels of relaxation.
- **Interpolation - no Relax** – If the policy value matches the parameter value exactly, return the factor. If the policy value falls between two parameters values, then the interpolated factor is proportionally between the two factors. Interpolation without relaxing requires that both bounds for the input parameter be found at the same level of relaxation.

For more information, see:

- “Rate table definition” on page 517
- “Matching a factor in the rate table” on page 521
- “Rate table with interpolated rate factor” on page 524
- *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 Matching Rule selected as part of the Add action. For a description of the matching rules, see “Parameters tab of Rate Table Definition screen” 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.
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.
Data Type	A drop-down list of the data types in the physical table and supported by the Matching Rule . If the Physical Table is DefaultRateFactorRow , the Exact Match choices are: String , Integer , Decimal , Boolean , and Date . The Range choices are: Integer , Decimal , and Date .
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 Decimal data type.
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 521.
Column Code	A name for the (logical) column. This field is visible only for range parameters. For other parameters, the value is not visible but is set to the same value as Code .
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. This field is visible only for range parameters. For other parameters, the value is not visible but is set to the same value as Label .
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 the <i>Configuration Guide</i> for more information.
Allow Multiple Values	Select Yes to allow multiple values in rate table content for this parameter. This field is selectable if all of the following are true: <ul style="list-style-type: none"> • You have selected a value provider. • The matching rule is exact match. • The data type is string. For more information, see “Rate table definition” on page 517.
Arguments	These are arguments to the value provider. For Typelist Value Provider , the Arguments are the list of typelists defined in PolicyCenter.

Field	Description
	<p>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 custom value providers in the default implementation take the following arguments:</p> <ul style="list-style-type: none"> • Coverage Value Provider – Requires no arguments. Any argument values are ignored. • Coverage Term Value Provider – A single optional argument. <ul style="list-style-type: none"> ◦ 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. • 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"> ◦ Coverage Code – Code of the coverage in product model (configured for the policy line associated with the rate table). ◦ Coverage Term Code – Code of the coverage term under the coverage. • Termless Coverage Value Provider – Very similar to Coverage value provider but find only those coverages that have no terms. • 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"> ◦ Rate Table Code – Code of a source rate table. ◦ Column Name – Name of the column in the source rate table.
Depends On	<p>Defines a dependency from one column of a rate table to another. 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:</p> <ul style="list-style-type: none"> • Specify a predetermined value provider argument that represents the coverage. (See the description for the Value Provider.) • 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. <p>A parameter cannot depend upon a parameter that allows multiple values.</p>

Factors tab of Rate Table Definition screen

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. The **Factors** tab also includes the **String**, **Integer**, **Decimal**, **Boolean**, and **Date** data types.

See also

- “Rate table with multiple factors” on page 523
- “Creating and using a rate table with a multiple factors” on page 585

Argument Sources tab of Rate Table Definition screen

In the **Rate Table Definition** screen, use the **Argument Sources** tab to specify one or more argument source sets. An argument source set specifies default argument sources for columns in the rate table definition. 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.

For each source set in the **Source** panel, the **Parameter Name** represents a column in the rate table definition. These columns (or parameters) are defined 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.

You cannot delete or edit an argument source set if it is used in a rate table lookup in a rate routine. Every rate table definition must have an argument source set.

For examples in the sample data, see the **BaseRate** and **DeductibleFactor** rate table definitions.

See also

- “Parameters tab of Rate Table Definition screen” on page 559 for descriptions of the **Value Provider Type** and **Value Providers** fields

Usage tab of Rate Table Definition screen

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 531 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 Rate Book screen for the selected rate book edition.
Rate Book Status	The status of the rate book.
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. Values are:</p> <ul style="list-style-type: none"> • Owned-Not Shared – This rate book is the first rate book to include this rate table. No other rate book includes this rate table. • Owned-Shared – This rate book is the first rate book to include this rate table, and another rate book includes this rate table. • Referencing – The rate book includes this rate table, but another rate book owns this rate table. <p>These values are the same as in the Usage column in the Rate Book screen on the Included Rate Tables tab. 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 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. Therefore, the custom table must have a foreign key attribute pointing to the **RateTable** entity. And the foreign key must be named **RateTable**.
- At least one parameter has been defined.

- At least one factor has been defined.
- At least one argument source set 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.
- If parameter's value provider is Typelist Value Provider, then arguments must have a value.

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 531 for more information.

Selecting a rate table

There are two ways to select a rate table for viewing or editing:

- From the Rate Book screen
- From the **Rate Table Definition**—**Usage** tab

Select a rate table from the Rate Book screen

Procedure

1. Select a rate book to view from the **Rate Book Search Results** pane to go to the Rate Book 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.

Select a rate table from the rate table definition editor Usage tab

Procedure

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
Name	Appears at the top of the screen. The rate table definition name in the default locale.
Code	The rate table definition code.
Description	The rate table definition description in the default locale.
Policy Line	The rate table definition policy line. When viewing the rate table, this field appears blank if the rate table applies to all policy lines.
In Rate Book	The rate book name in the default locale.

Field	Description
Usage	<p>Displays Referencing, Owned-Shared, or Owned-Not Shared to describe the relationship between the rate table version and the rate book.</p> <p>Values are:</p> <ul style="list-style-type: none"> • Owned-Not Shared – This rate book is the first rate book to include this rate table, and no other rate book includes this rate table. • Owned-Shared – This rate book is the first rate book to include this rate table, and another rate book includes this rate table. • Referencing – The rate book includes this rate table, but another rate book owns this rate table.

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 screen” on page 559 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 “Edit rate table content in PolicyCenter” on page 564.

You can also maintain the rate table content in Microsoft Excel. For more information, see “Edit rate table content in Excel” on page 564.

Edit rate table content in PolicyCenter

About this task

You can edit rate table content only if the rate table is included in rate books with **Draft** status.

Procedure

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 screen.
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.

Edit rate table content in Excel

Procedure

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 screen.
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 hypertext link 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">• For Range Max Excluded – Less than the value of the max column.• For Range Max Included – Less than or equal to the value of the max column.
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

- “Edit rate table content in PolicyCenter” on page 564
- “Edit 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

- “Edit rate table content in PolicyCenter” on page 564
- “Edit rate table content in Excel” on page 564

Working with rate routines

In PolicyCenter, you can create and edit rate routines.

Access rate routines

Before you begin

You must have the **ratebookview** permission to access rate routines.

Procedure

1. In PolicyCenter, select **Administration**→**Rating**→**Rate Routines**.

This screen has **Policy Line**, **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, select **<not specified>** for **Policy Line** and leave **Code** and **Name** blank.
- To view rate routines that apply to all policy lines, select **<applies to all>** and leave **Code** and **Name** blank. In the search results, **Policy Line** set to **<applies to all>** appears blank.

2. Enter your search criteria, then click **Search**.

For each search result, PolicyCenter displays the **Name**, **Code**, and **Version**.

Add new rate routine

Before you begin

You must have the **ratebookview** and **ratebookedit** permissions to add a new rate routine.

About this task

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.

After adding the rate routine in PolicyCenter, add the rate routine to a rate book and to the rating plugin as described in “Add new rate routine” on page 566 and “Add new rate routine” on page 566.

Procedure

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
Code	A unique code for the routine. This is the <code>Code</code> on the <code>CalcRoutineDefinition</code> entity instance.
Name	A name for the routine. This is the <code>Name</code> on the <code>CalcRoutineDefinition</code> entity instance.
Jurisdiction	The jurisdiction to which this routine applies. Select <code><applies to all></code> if the rate routine can apply to all jurisdictions.
Version	Read-only. A version for the routine. This is the <code>Version</code> on the <code>CalcRoutineDefinition</code> entity instance.
Description	A description for the routine. This is the <code>Description</code> on the <code>CalcRoutineDefinition</code> entity instance.
Policy Line	The policy line that this routine applies to. Select <code><applies to all></code> if the rate routine can apply to all policy lines.
Parameter Set	Select a parameter set. The drop-down list displays the parameter sets defined for the selected Line of Business .
Parameters in Parameter Set	Read-only. The parameters in the selected Parameter Set . If you hover the mouse over the parameters, PolicyCenter displays the value in the <code>Type</code> column for each parameter. For more information, see “Alter parameters in parameter set in Rating Management” on page 580.
Last Updated By	The name of the user who last updated this rate routine. This field is read-only.
Last Update Time	The time that the user last updated this rate routine. This field is read-only.

When viewing the Rate Routine screen, **Policy Line** and **Jurisdiction** set to `<applies to all>` appear blank.

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.
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 Book screen” on page 549.
7. Add the rate routine to the rating engine for the line of business. See the *Configuration Guide*.

Delete rate routine

Before you begin

You must have the `ratebookview` and `ratebookedit` permissions to delete a rate routine.

Procedure

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 a rate book with **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 528.
- **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 528.
- **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 553

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 Add 1 Row or Add 10 Rows 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"> • Duplicate – 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. • Insert Before – Inserts a blank step before each selected step. • Insert After – Inserts a blank step after each selected step.
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.
Error#	Displays the error number. The error description appears in Validation Results . Appears in edit mode only.
Instruction	Optionally, select a value. The instruction field is the target of an assignment operator, a conditional, or a section comment. Select Clear to remove the value of this field. If the instruction field is the target of an assignment operator, you can set the Instruction field to one of the following: <ul style="list-style-type: none"> • Properties on the cost • Parameters, including rate modifiers • Variables For more information, see “Instruction and operand types in rate routine steps” on page 571. 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 527.
Op	The operator in a rate routine step can be one of the following: <ul style="list-style-type: none"> • Blank • Assignment – Assign the operand to the Instruction field. • Arithmetic operators – Includes multiply, divide, add, and subtract. • 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. 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.
Operand	Select one of the following operand types: <ul style="list-style-type: none"> • Properties on the cost • Functions • Parameters • Rate tables • Variables • Conditional expressions • Constants • Date constants • Typelist values • Scale For more information, see “Instruction and operand types in rate routine steps” on page 571. 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 527.
)	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:

Operator	Description
x	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:

Operator	Description
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.

Operator	Scale	Value	Rounded value	Description
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 rate routine steps” on page 576 in “Instruction and operand types in rate routine steps” on page 571
- “Fields in a rate routine step” on page 568
- Configuration Guide*

Instruction and operand types in rate routine steps

In rate routine steps, 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	Functions	Not specified	<ul style="list-style-type: none"> Instruction – void (function has no return value) Operand – Any type
	Rate tables	table:	String, integer, decimal, Boolean, or date
	Constants		Numeric, Boolean, string, or null
	Date constants		Date
	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 527.

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 the *Configuration Guide*.

Do not use a policy entity the **Instruction** target in a rate routine step

Do not use a entity that is part of the policy as the target of the **Instruction** column on the left side of a rate routine step. This requires that PolicyCenter update the policy branch. This is an unexpected use of rate routines. Rate routines are for calculating costs for a policy. They are not expected to make changes to the policy branch.

For example, in a parameter set you have a writable **Vehicle** parameter defined as `entity.VehicleDriver` type. This entity has a **PolicyDriver** property that is a foreign key to a **PolicyDriver** entity instance. In a rate routine that uses this parameter set, avoid specifying `Vehicle.PolicyDriver` as the target of an **Instruction**.

Drop-down lists for the instruction and operand fields in rate routine steps

The **Instruction** drop-down list contains properties or local variables that are writable. The list can also contain functions that return void.

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 in rate routine steps

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 527.

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 “Cost delegate” on page 394 and “In rate routine, specify coverage as flat-rated” on page 578.
- **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 411 and the *Integration Guide*.

Parameters in rate routine steps

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 can also select writable 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.

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 581.

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**.

Filter parameter set fields of same type when entering parameters in rate routine

About this task

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.

Procedure

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.

Variables in rate routine steps

If rate routine steps, 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 in rate routine steps

In rate routine steps, 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 in rate routine steps

In rate routine steps, 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 operator	Description
<	Less than
=	Equal
≠	Not equal
>	Greater than
≤	Less than or equal to
≥	Greater than or equal
IN	The left operand is a member of the set defined by the right operand. You can choose this operator if the left operand is a typekey value. The right operand choices are limited to a single typekey or a list of typekeys in a typelist. Both left and right operand can be parameter data, rate table lookup, function, local variable. Left operand can also be a single typekey constant. Right operand can also be a list of constants.
NOT IN	The left operand is not a member of the right operand set. You can choose this operator if the left operand is a typekey value. The right operand choices are limited to a single typekey or a list of typekeys in a typelist. Both left and right operand can be parameter data, rate table lookup, function, local variable. Left operand can also be a single typekey constant. Right operand can also be a list of constants.

An example of a simple conditional expression is:

```
AdjustedRate ≤ BaseRate
```

You can create a complex conditional expression by combining two or more conditional expressions separated by **AND** or **OR**. For example:

```
BaseRate ≠ AdjustedRate  
AND BaseRate > 10
```

You can add parentheses to the conditional expression. The parentheses must be balanced. For example:

```
( BaseRate ≠ AdjustedRate )  
AND ( BaseRate > 10 )
```

Functions in rate routine steps

In rate routine steps, functions are available in operands. Functions that return void are available in instructions. 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.

See also

- *Configuration Guide*

Rate tables in rate routine steps

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.

Select rate table in rate routine step

About this task

You can select a rate table and return value.

Procedure

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**.

You can select rate tables that match the rate routine's line of business.

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 is **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. This step is required if the rate table definition has not specified a default argument source.

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 585.

Constants in rate routine steps

In rate routine steps, 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 in rate routine steps

In rate routine steps, 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 in rate routine steps

In rate routine steps, 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 in rate routine steps

In rate routine steps, 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 “Operators in rate routine steps” on page 569 in “Operators in rate routine steps” on page 569.

Section comment in rate routine steps

In rate routine steps, 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.

Specify function as operand in rate routine step

Procedure

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

- *Configuration Guide*

Create rate routine for another jurisdiction

About this task

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 528.

You can extend PolicyCenter to create variant identifiers for other features of the policy such as underwriting company. For more information, see the *Configuration Guide*.

Procedure

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 “Add new rate book” on page 548 and “Rate Book screen” on page 549.

In rate routine, specify coverage as flat-rated

About this task

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 “Cost delegate” on page 394.

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.

Procedure

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 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.

Editing long rate routines

When editing a long rate routine, you can edit the rate routine by section. A long rate routine has more steps than the value of the indexing threshold, the **RateRoutineIndexingThreshold** parameter in **config.xml**. In the base configuration, this parameter is set to 150.

When you edit a long rate routine, PolicyCenter displays a **View** drop-down list containing entries for each section. Select **Show All** from the **View** drop-down list to view the entire rate routine. Select a section, and PolicyCenter begins the display at that section and succeeding rows up to the indexing threshold. If a single section is longer than the indexing threshold, PolicyCenter breaks that section evenly into multiple entries on the **View** drop-down list.

If you make any edit to the current Rate Routine, PolicyCenter displays a warning message that you cannot change views when there are pending changes. You must update or cancel before changing views. When there are pending changes, only the current section is available for editing.

When editing a long rate routine, the **Append** button is not available.

After clicking **Validate**, validation error numbers appear after each section under the **View** button.

View rating worksheets

Before you begin

The user must have the **View rating worksheet** permission to view a rating worksheet. The code for this permission is `ratingworksheetview`.

Procedure

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 table name.
- Rate routine code and version.

If cascaded lookup causes the rate table to be in a different rate book than the rate routine, then this field also displays the rate book name and edition.

- 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 **Expand All** to view all rows of the rating worksheet. This button does not appear if the number of rows is greater than 10,000. Use **Download** to view rating worksheets larger than 10,000 rows.

5. 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**.

6. Click **Download** to export the rating worksheet to CSV or HTML5 format.

See also

- “Rating worksheets in Rating Management” on page 542
- *Configuration Guide*

Working with parameter sets in Rating Management

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.

See also

- “Parameter sets in Rating Management” on page 530

Access parameter sets in Rating Management

Procedure

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.

Add parameter set in Rating Management

Procedure

1. On the **Parameter Sets** screen, click **Add**.

PolicyCenter adds a blank parameter set to the list of parameter sets.

2. Enter values for **Code** and **Name**.

3. Enter a value for **Policy Line**. Select **<applies to all>** if this parameter set can apply to all policy lines.

When viewing a parameter set, **Policy Line** set to **<applies to all>** appears blank.

4. Select **Include Cost** if the parameter set will be used in rate routines that calculate properties on the cost, such as the base rate, adjusted rate, and term amount.

Alter parameters in parameter set in Rating Management

About this task

You can add, edit, or remove parameters in a parameter set.

Procedure

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 is disabled when a rate book with status other than **Draft** references the parameter set through an included rate table definition or rate routine.

4. In the **Parameters** tab, click **Add** to add a parameter.

5. Specify the following information for the parameter:

Field	Description
Name	Select a name from the drop-down list. The names are specified in Studio.
Type	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.
Use Wrapper	Select this option to use a wrapper that selects a coverage based on characteristics of the policy. See “Combine similar parameter sets with wrappers” on page 531.
Wrapper/Coverage	If the Type specifies an entity that is a coverage, then you can select a specific coverage in the policy line. For example, a parameter with Type set to entity.Coverage specifies an entity that is a coverage.
Code	If you selected Use Wrapper , then click the Search icon  to select a coverage wrapper.

Field	Description
Writable	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 Instruction field of a step.

In some cases, you must 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, 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 rate routine access to coverage terms, options, and packages in policy data. Define the parameter with the **Type** field specifying `entity.Coverage`, and the **Coverage** field specifying a coverage pattern code. For more information, see “Parameters in rate routine steps” 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

- “Filter parameter set fields of same type when entering parameters in rate routine” on page 573
- *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.LobLine` as a fully qualified object type. For example, enter `entity.CommercialPropertyLine` for the commercial property line.

See also

- “Parameters in rate routine steps” on page 573

Working with impact testing in Rating Management

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.

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** role has 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.

Test periods generated by impact testing

You can use impact testing to generate test policy periods rated using the current rate books and rate books that you select.

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.

Generate test periods in impact testing

Procedure

1. In PolicyCenter, navigate to **Administration**→**Rating**→**Impact Testing**.
PolicyCenter displays the **Choose Policies** screen.
2. Do one of the following:
 - To continue a previously started test, click **Next with Previous Test Case** to advance the wizard. Go to “Generate test periods in impact testing” on page 582.
 - Specify criteria to filter policies for testing. None of these fields is required.

Field	Description
Products	Choose one or more products.

Field	Description
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"> • Specify codes – One or more postal codes separated by commas. • 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.
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.

3. Click **Search** to display the list of policies that meet the search criteria.
 4. Repeat specifying criteria in “Generate test periods in impact testing” on page 582 and search in “Generate test periods in impact testing” on page 582 until you are satisfied with the search results.
 5. 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.
6. Depending upon the button that you clicked in “Generate test periods in impact testing” on page 582, 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 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 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 “Generate test periods in impact testing” on page 582.
- 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

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 523

Create rate table with multiple rate factors

About this task

In this topic, you add two additional rate factors to a copy of the **Base Rate** rate table.

Procedure

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

Field	Value
Priority	20
Column Code	Boolean Factor
Column Label	Boolean Factor
Display Type	Normal
Physical Column	bit1

8. Click **Update** to save your changes.

Next steps

For instructions on using the multiple factor rate table in a rate routine, see “Using the multiple factor rate table in a rate routine” on page 586.

Using the multiple factor rate table in a rate routine

Before you begin

For instructions on creating a multiple factor rate table, see “Create rate table with multiple rate factors” on page 585.

Procedure

1. In PolicyCenter, select **Administration**→**Rating**→**Rate Routines**.
2. On the **Rate Routines** screen, click **Add** to create a new rate routine.
3. 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.

4. In the **Instruction** field of the first step, select **New**→**New Variable** and name it **varMultiFactor**.
5. Select \leftarrow as the **Op**.
6. In the **Operand**, select **Rate Table**, and select the **Base Rate with Multiple Factors** rate table.
7. For **Return Value**, select **[All]**.
8. 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.

Adding parameters to an in-use rate table definition

After a rate table definition is used in a rate book, you can add and delete parameters from the rate table definition. The types of changes you can make depend upon several factors including whether the rate table definition is used in an active rate book. This topic provides step-by-step instructions for several cases.

See also

- “Changes to parameters in rate table definitions” on page 533

Add parameters to rate table definition used in active rate book

About this task

This example uses the Generic RTM Demo Rating rate book from the sample data. This is an active rate book. This rate book includes the GenericStateTax rate table and the Generic State Tax Calculation rate routine. Because the rate book is active, you can only add a parameter with higher priority than existing parameters. To use the parameter, you must also add a new argument source set.

To add the parameter to the rate table definition:

Procedure

1. Navigate to the rate book, Generic RTM Demo Rating.
The status is active.
2. Under the **Included Rate Tables** tab, click to view the rate table, GenericStateTax.
The rate table content contains one parameter column for Jurisdiction.
3. Navigate to the rate table definition of GenericStateTax and edit it.
4. On the **Parameters** tab, **Add** a new parameter named **NewParamHighPriority**.
5. Fill in all required fields, and set the priority to 20, higher than parameters already defined.
6. On the **Argument Sources** tab, click **Add Source Set**.
The source set contains both parameters.
7. Specify a name and code for the source set.
8. Click **Update** to save your changes.

PolicyCenter displays an error message if the priority of the parameter is not high enough.

Next steps

Now that you have added the parameter, you can view it in the active rate book. See “View the parameter in the active rate book” on page 587.

View the parameter in the active rate book

Before you begin

Before you can view the parameter, you must have added a parameter to an rate table definition. See “Add parameters to rate table definition used in active rate book” on page 587.

About this task

This example uses the Generic RTM Demo Rating rate book from the sample data.

Procedure

1. Navigate to the active rate book, Generic RTM Demo Rating.
2. Under the **Included Rate Tables** tab, click to view the rate table, GenericStateTax.

The rate table content contains a column for the new parameter, NewParamHighPriority. The column is empty and you cannot add values because the rate book status is active.

Add the parameter in a new rate book edition

Before you begin

Before you can add the parameter to a new rate book edition, you must first add the parameter. See “Add parameters to rate table definition used in active rate book” on page 587.

About this task

This example uses the Generic RTM Demo Rating rate book from the sample data.

Procedure

1. In the active rate book, Generic RTM Demo Rating, create a new edition. Notice that the new edition is in draft status.
2. View the GenericStateTax rate table and edit the rate table content.
You can set values for the new parameter.

Add parameter to rate table used in rate routine

About this task

You can add parameters to a rate table used in rate routines. You can also add parameters if that rate routine is included in a rate book. The rate book cannot be in active status.

First, add parameters to the rate routine.

Procedure

1. Create a new rate table definition by copying YouthfulDriverAge and naming the copy YouthfulDriverAgeNotInRB.
2. Make a copy of the PA Assign Driver Style 1 rate routine. Name this rate routine PA Assign Driver RT in RR with code `pa_assign_driver_RT_in_RR`.
3. In the rate routine step with the operand table: YouthfulDriverAge, select Rate Table and change the rate table to YouthfulDriverAgeNotInRB.
4. Add a new parameter to the rate table definition. This parameter can have any priority.
5. On the **Argument Sources** tab, specify an argument source.
6. Save the rate table definition.
7. Go to the PA Assign Driver RT in RB rate routine.
Because of the new parameter, PolicyCenter displays a message that it is unable to reconcile the rate table definition.
8. Edit the rate routine.
9. In the rate routine step with the operand table: YouthfulDriverAgeNotInRB, select Rate Table and click **OK** to confirm the addition of the new parameter. This refreshes the rate table lookup.
In the following steps, you add the rate routine to a rate book.
10. Navigate to the Rate Books screen and add a new rate book named RR in RB, edition 1.
11. Set the policy to Personal Auto Line.
12. Add the `pa_assign_driver_RT_in_RR` rate routine to the rate book.
PolicyCenter displays a warning that the rate table does not exist in this rate book. You can ignore this warning.
13. Click **Update** to save your changes.

14. Promote the rate book to stage status.
15. Edit the rate table definition for YouthfulDriverAgeNotInRB.
16. Add a new parameter called NewParamPromotedRB with priority 3.
17. Go to the **Argument Sources** tab.
The new parameter is not in the source set.
18. Add an argument source set named NewParamPromotedRB.
The new parameter is automatically included.
19. Specify argument sources for all parameters.
Now, you can use the new parameter in a rate routine.
20. Return the RR in RB rate book to draft status.
21. Edit the pa_assign_driver_RT_in_RR rate routine.
22. Refresh the rate table lookup in the rate routine by selecting the rate table.
23. Change the argument source set to SetParamPromotedRB and click **OK**.

Part 7

Reinsurance Management

Reinsurance Management concepts

Reinsurance is insurance risk transferred to another insurance company for all or part of an assumed liability. In other words, reinsurance is insurance for insurance companies. When a company reinsures 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.

The insurance company directly selling the policy is also known in the industry as the *insurer*, the *reinsured*, or the *ceding company*. The Guidewire term for this company that directly sells the policy is *insurer*. An insurance company accepting ceded risks is known as the *reinsurer*.

Guidewire Reinsurance Management provides reinsurance for all lines of business.

Note: 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.

See also

- “Multicurrency and reinsurance” on page 491

Reinsurance program basics

Insurers 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 insurer and the reinsurer to provide coverage for all risks of a certain type.

Insurers group reinsurance treaties into reinsurance programs to cover policy risks in a consistent way that meets the insurer's business goals. Insurers also group treaties into programs to ensure that they have no gaps in coverage and to ensure that they do not duplicate coverage.

An insurer typically operates several reinsurance programs. The insurer 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 insurer'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 insurer 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

An insurer 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 insurer 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 insurer might find it advantageous to bundle various types of reinsurance in a way that maximizes its ability to achieve these business goals.

For instance:

- Insurers 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 an insurer can be free of fear of multiple large losses, it can comfortably take on more risk.
- Insurers that seek to stabilize their net income flow benefit from reinsurance that takes a percent of the loss above a certain point.
- Insurers 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 an insurer has one or more of these business goals in mind, common industry practice has established that the insurer can achieve these goals through reinsurance. In setting up reinsurance programs, insurers take into account factors such as:

- The insurer's average policy claim losses and premium intake
- Likelihood of catastrophe
- Proximity of policies taken out in a geographic location

Insurers 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 insurer 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.

Insurers 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 insurer's reinsurance programs, the insurer can take out a facultative reinsurance agreement for that particular risk. Often, the insurer 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 insurer typically has one active reinsurance program for policy risks falling in the property reinsurance coverage group. The insurer typically also has another active reinsurance program for policy risks falling in the liability reinsurance coverage group. The insurer might also have one future and multiple past programs for each of these reinsurance coverage groups. While there is some consistency among insurers, a insurer may have their own way of categorizing the risks in various lines into different 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 599

[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 insurer 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 598.
- “Proportional agreements” on page 599.
- “Non-proportional agreements” on page 602.

Using facultative agreements in addition to programs

After an insurer 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 insurer can add facultative reinsurance agreements to cover the uncovered portion of each very high risk.

For more information, see “Facultative agreements” on page 599.

Reinsurance program example

The reinsurance program example in this topic illustrates the reinsurance program concepts explained in “Reinsurance program design” on page 594.

In our example, the reinsurance insurer 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.

Reinsurance program example: risk and loss in treaty composition

Property reinsurance program

The insurer’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 insurer to take in more business below \$5 million than it had in the past. The insurer can take in more business because the insurer shares the risk of loss payments with two other reinsurers under Treaty One and Treaty Two.

Treaty Three enables the insurer to expand into a higher risk, more lucrative market. Before adding Treaty Three, the insurer could not afford the risk of insuring losses above \$5 million. The insurer finds it attractive to include Treaty Three in its property program because losses above \$5 million occur infrequently. The insurer 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 insurer 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 insurer 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.

Reinsurance program example: 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 insurer'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 "Reinsurance program example: risk and loss in treaty composition" on page 596.

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 reinsure. 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. The proportional share is based on the total insured value, not the amount of a specific loss. In general, a proportional agreement gets x% of the gross net premium to cover x% of the gross net losses, no matter the size of the loss. Detailed examples of proportion determination are provided in "Proportional treaties" on page 599.

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.

Reinsurance program example: attaching policies to reinsurance programs and treaties

This example draws upon the property example "Reinsurance program example: risks and coverables when applying program treaties to a loss" on page 597. In that example, Treaty One and Treaty Two share with the insurer in any loss beginning from zero on any risk that falls within their covered risk amount. For example, the insurer recovers from both Treaty One and Treaty Two on any property risk with a total insured value of \$1.5 million.

The insurer 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 insurer can recover only up to the smaller of the total insured value or \$10 million.

In short, all of the insurer'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 Reinsurance Management links reinsurance to policies” on page 607

Reinsurance agreements

There are two kinds of reinsurance agreements, treaties and facultative agreements.

- **Treaty** – An agreement between the insurer 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 insurer 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 on by the insurer and reinsurer.

Treaties

A treaty is an agreement between the insurer and the reinsurer that provides reinsurance without the insurer 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 insurer cedes to the reinsurer a portion of each risk that the treaty covers.

For example, the insurer 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 599
- “Non-proportional treaties” on page 603

Reinsurance coverage groups for treaties

You can group individual coverages into a reinsurance coverage group. Treaties are written to cover losses 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.

Some insurers have reinsurance agreements that provide broad terms for coverage that do not explicitly specify the risks. For example, an insurer has a reinsurance agreement that can be attached to a class of risks without further negotiation with the reinsurer. In PolicyCenter, you can model this agreement as a treaty that attaches to a class of risks based on business logic or is included by a user in a program.

For a specific risk, the insurer and the reinsurer each have free choice in arranging the reinsurance. The insurer 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 insurer 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

- “Facultative agreements” on page 599
- “Non-proportional facultative agreements” on page 605

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, an insurer 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.

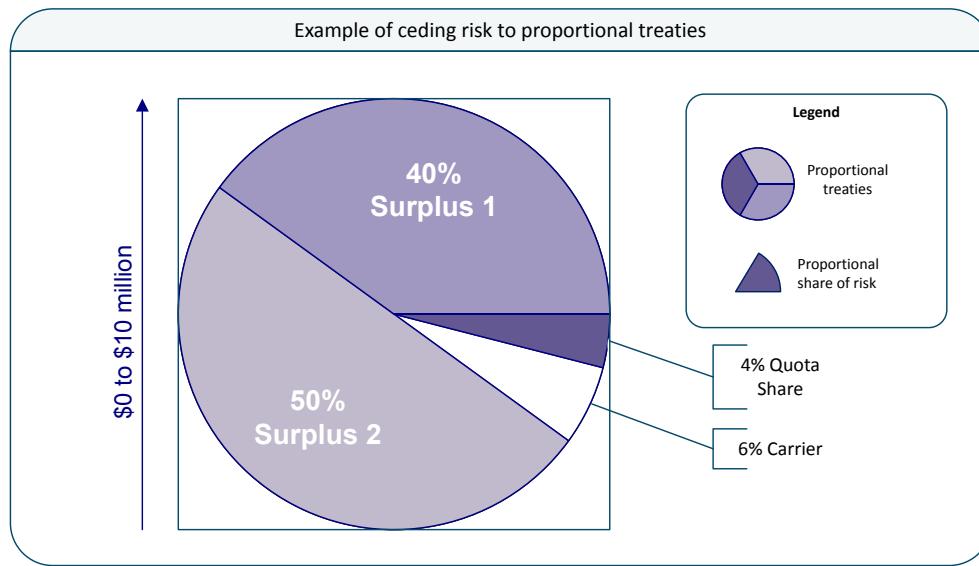
- **Surplus** – The surplus treaty provides reinsurance coverage from a starting value up to the coverage limit. The way in which the percentage of premium is ceded and losses are paid is similar to quota share.

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%
Quota share	From \$0 to \$1 million ceding 40% of the risk to the reinsurer	\$400,000	\$400,000 of \$10 million = 4%
Insurer's share	From \$0 to \$1 million 60% of the risk retained by the insurer	\$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
Insurer'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
Insurer'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 insurer 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, you might think of the agreement as representing 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

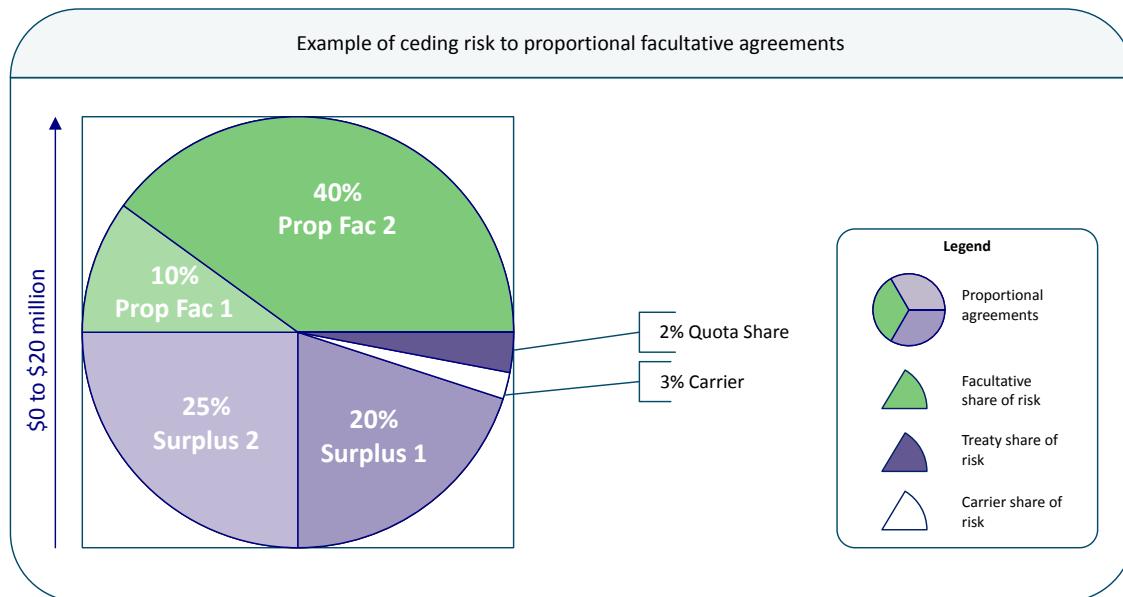
This example shows how risk is ceded in a proportional facultative agreement with a monetary amount entered in the **Amount of Risk Ceded** field on the **Facultative** screen. This example builds upon Example of ceding risk to proportional treaties which provided reinsurance from \$0 to 10 million on a specific risk. However, in this example the risk is valued at \$20 million. The insurer 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
Proportional Facultative Agreements			
Proportional Facultative 2		\$8 million	\$8 million of \$20 million = 40%
Proportional Facultative 1		\$2 million	\$2 million of \$20 million = 10%
Proportional treaties			
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%

Treaty	Layers of reinsurance	Amount of risk ceded	Proportional share of risk
Insurer's share	From \$0 to \$1 million 60% of the risk retained by the insurer	\$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
Proportional facultative agreements			
Proportional facultative 2	40% of loss amount	\$8 million	\$2 million
Proportional facultative 1	10% of loss amount	\$2 million	\$500,000
Proportional treaties			
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
Insurer'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 insurer is responsible for 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 insurer 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 insurer 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, but based on a time period rather than an event. 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 “Non-proportional treaties” on page 603.

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% insurer 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 insurer provides 100% coverage.	
Excess of Loss	Attachment point \$1 million Coverage limit \$3 million
From \$0 to \$1 million, the insurer provides 100% coverage.	

Losses would be covered by this agreement as follows:

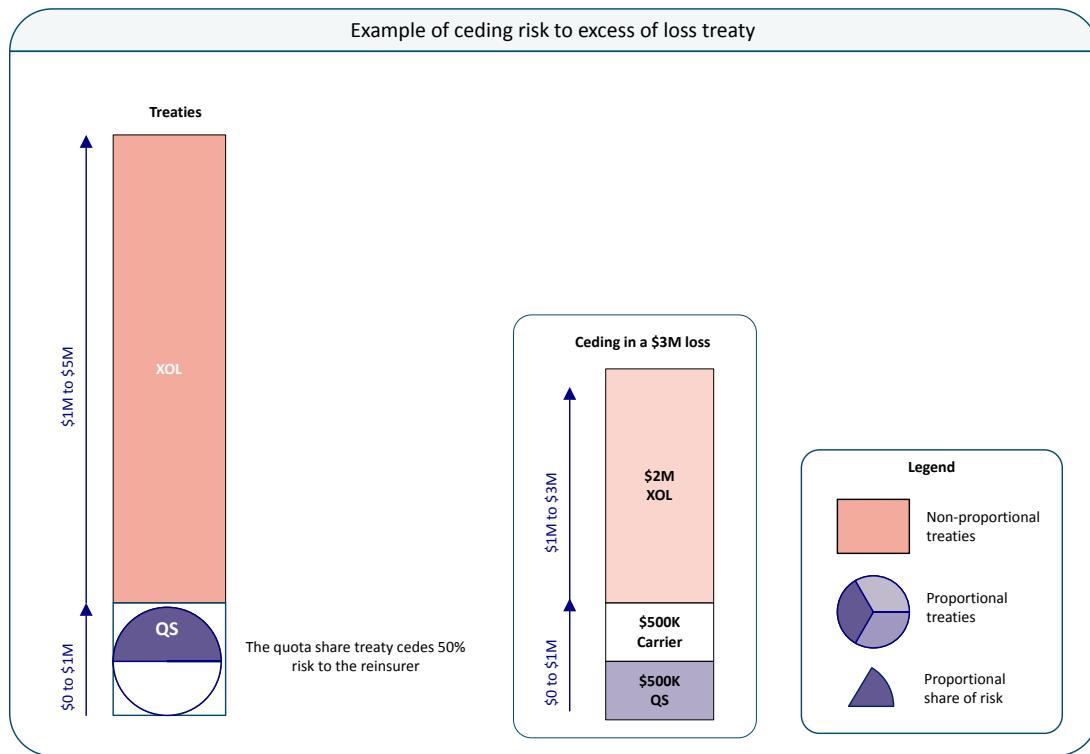
- \$900,000 loss – The reinsurer pays nothing because it is under the \$1 million attachment point.
- \$2,500,000 loss – The insurer pays the first \$1 million, and the reinsurer pays the next \$1,500,000.
- \$4,500,000 loss – The insurer pays the first \$1 million. The reinsurer pays the next \$2 million up to the reinsurance limit of \$3 million. The insurer pays the last \$1.5 million, unless the insurer 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 insurer 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 insurer 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 insurer'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

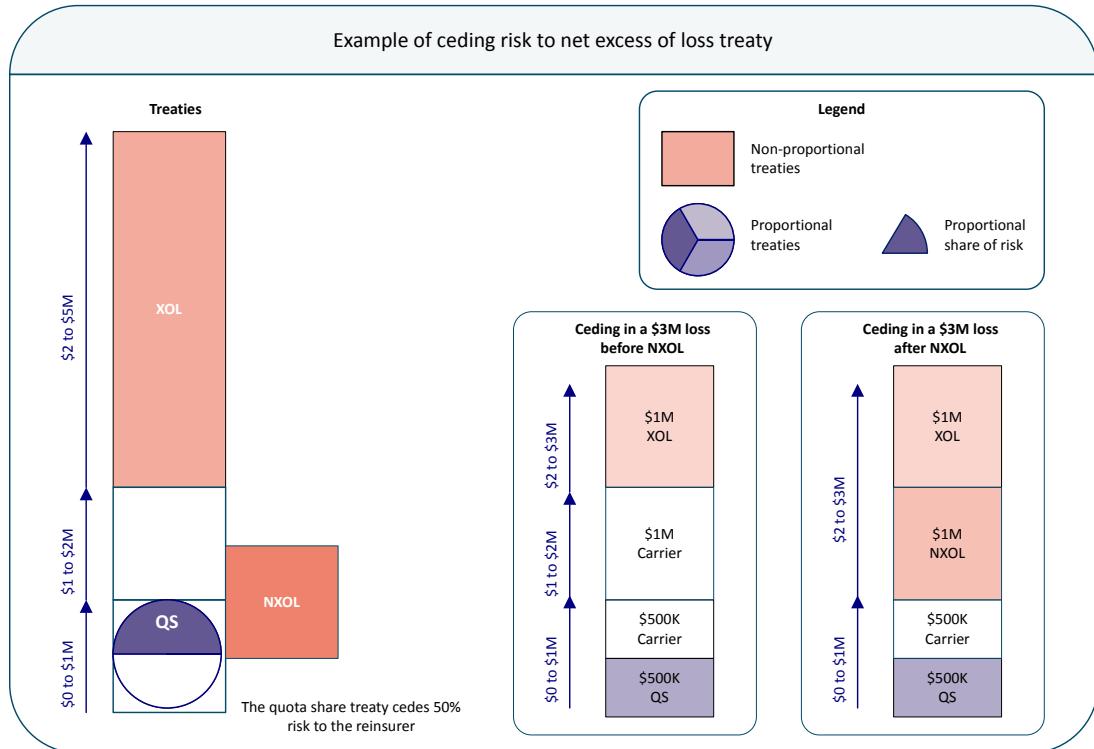
The insurer has a program that contains three treaties:

- 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 insurer 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 insurer's net loss is \$1.5 million, but the insurer collects \$1 million from the net excess of loss agreement for the amount of net loss above \$500,000. The insurer'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

An insurer 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 insurer can have a per event agreement to provide coverage for \$100 million in excess of \$20 million. The insurer retains \$20 million in aggregate net risk. The insurer 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 insurer had planned for 2,000 slip and fall liability losses in a year, but there are claims for 10,000. The insurer's risk retention is unacceptably high. To protect against this eventuality, the insurer 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 insurer 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 insurer'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 use 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
Non-proportional						
Annual Aggregate	●			●		●
Per Event	●			●		●
Excess of Loss	●		●		●	●
Net Excess of Loss	●		●		●	●
Facultative Excess of Loss		●	●		●	
Facultative Net Excess of Loss		●	●		●	
Proportional						
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 Reinsurance Management attaches agreements to policies” on page 607

How Reinsurance Management links reinsurance to policies

Guidewire Reinsurance Management applies one or more reinsurance programs to a policy in PolicyCenter. PolicyCenter links the policy to each agreement in the program.

How Reinsurance Management attaches programs to policies

In a PolicyCenter policy transaction such as a submission, Reinsurance Management selects reinsurance programs to best meet the needs of the insurer, based on their business logic. This generally includes:

- Providing reinsurance over the term of the policy, even across program years.
- Providing coverage for each reinsurance coverage group of coverages on the policy.
- Selecting programs and agreements in the same currency as the TIV/SI of the risks.

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 links that location to a reinsurance program for each coverage group. PolicyCenter automatically manages reinsurance on policies that span program-year boundaries; no policy change is required.

See also

- “Multicurrency and reinsurance” on page 491

Invalid draft reinsurance programs can cause problems

In PolicyCenter, you can create and save draft reinsurance programs that do not pass validation required to make the program active. You can save draft programs that have issues such as inconsistent dates, inappropriate retention levels, inconsistent reinsurance coverage groups, and inconsistent currencies. In your implementation, you can use validation to ensure that active programs meet business and regulatory requirements.

There is only a potential problem if a draft program gets attached to a policy that is then bound or issued. In the base configuration, PolicyCenter attaches an active program first. However, PolicyCenter attaches 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 Reinsurance Management attaches agreements to policies

With Reinsurance Management, 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
Non-proportional treaties		
Annual Aggregate Treaty		•
Per Event Treaty		•
Excess of Loss Treaty	•	•
Net Excess of Loss Treaty	•	•
Non-proportional facultative agreements		
Facultative Excess of Loss	•	
Facultative Net Excess of Loss	•	
Proportional treaties		
Quota Share	•	
Surplus	•	
Proportional facultative agreements		
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 491

How Reinsurance Management selects a projected program if active program is unavailable

At a particular point in time, the reinsurance 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.

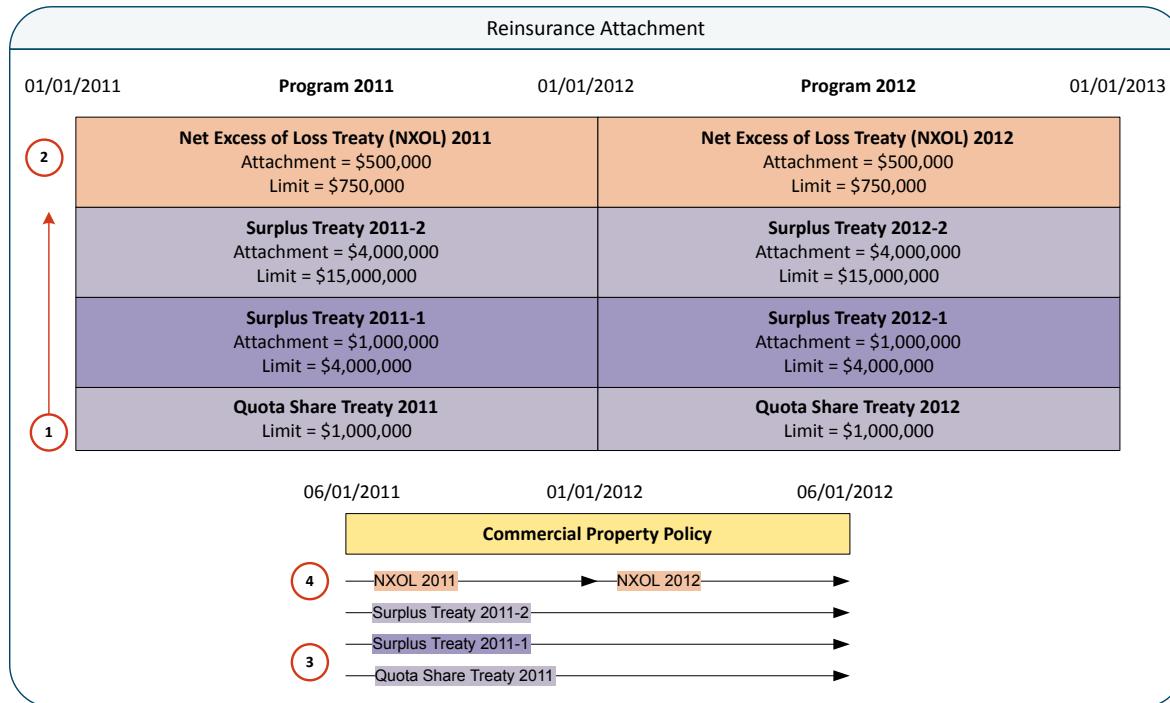
In the default configuration, the implementation of the **RIProgramFinder** interface selects the program. In the default configuration, the **RIProgramFinderImpl** class implements this interface.

See also

- *Integration Guide*

Example: policy attachment in Reinsurance Management

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: In the base configuration, a midterm policy change does not change the set of treaties that apply to a risk. There is no change because only the policy period effective date matters for policy attachment treaties. Only the effective date of the program matters for loss attachment treaties. You can configure variations of these two attachment methods.

See also

- “Add reinsurance to a policy” on page 620
- *Integration Guide*

Calculating ceded premiums in Reinsurance Management

In a reinsurance agreement, the reinsurer takes responsibility for part of the risk that the insurer assumed from the insured. In compensation, the insurer 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 insurer calculates how much of the direct premium is ceded to each agreement.

IMPORTANT In the base configuration, 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 the *Integration Guide*.

Premium ceding in a proportional treaty

For proportional treaties, the insurer cedes a share percentage of the premium for the total risk.

Assume a \$10-million risk as the basis for the following examples:

- The insurer retains all risk up to \$1 million, and there is a surplus treaty in place from \$1 million to \$5 million. The surplus treaty is ceded \$4 million of the \$10 million of risk (40%), and is ceded 40% of the *applicable premium*.
- The first million of risk is covered by a 80% quota share treaty (rather than being retained by the insurer). That quota share is ceded 80% of \$1 million of the \$10 million of risk, 80% of 10%, or 8% of the applicable premium.

The *applicable premium* is the gross net premium, or GNP. In many cases the GNP is not the total premium for the policy. One common situation is where there is an excess of loss agreement in place that limits the risk exposure of the proportional agreement participants.

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 insurer 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 insurer 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 491.

Calculating ceded premiums in Reinsurance Management

This topic describes how PolicyCenter calculates ceded premiums.

Review the following topics to understand the process flow.

Ceding premium to excess of loss agreements

If there are excess of loss (XOL) agreements with flat ceding amounts, cede those amounts. Cede any markup on these agreements. Calculate the gross net premium (GNP) as follows:

$$\text{GNP} = \text{Written Premium} - \text{XOL Ceded Premium} - \text{XOL Markup}$$

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 insurer’s net risk after deducting risk ceded to proportional agreements.

The process continues and calculates the ceded premiums for proportional risks. See “Ceding premium to proportional agreements” on page 611

Ceding 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 insurer the proportional agreements. The modified TIV is the TIV minus any amounts that any excess of loss treaties or agreements cover. Net excess of loss and aggregate treaties are not deducted because they apply only to the insurer’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:

Proportional Share of Risk = Amount of Risk Ceded / 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:

Prop Ceded Premium = GNP * Proportional Share of Risk

The process continues and cedes the premium to the facultative net excess of loss agreements. See “Ceding premium to facultative net excess of loss agreements” on page 612.

Ceding 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.

The process continues and cedes the premium to other treaties.

See also

- “View ceded premiums” on page 622
- Integration Guide*

Example: ceded premiums in Reinsurance Management

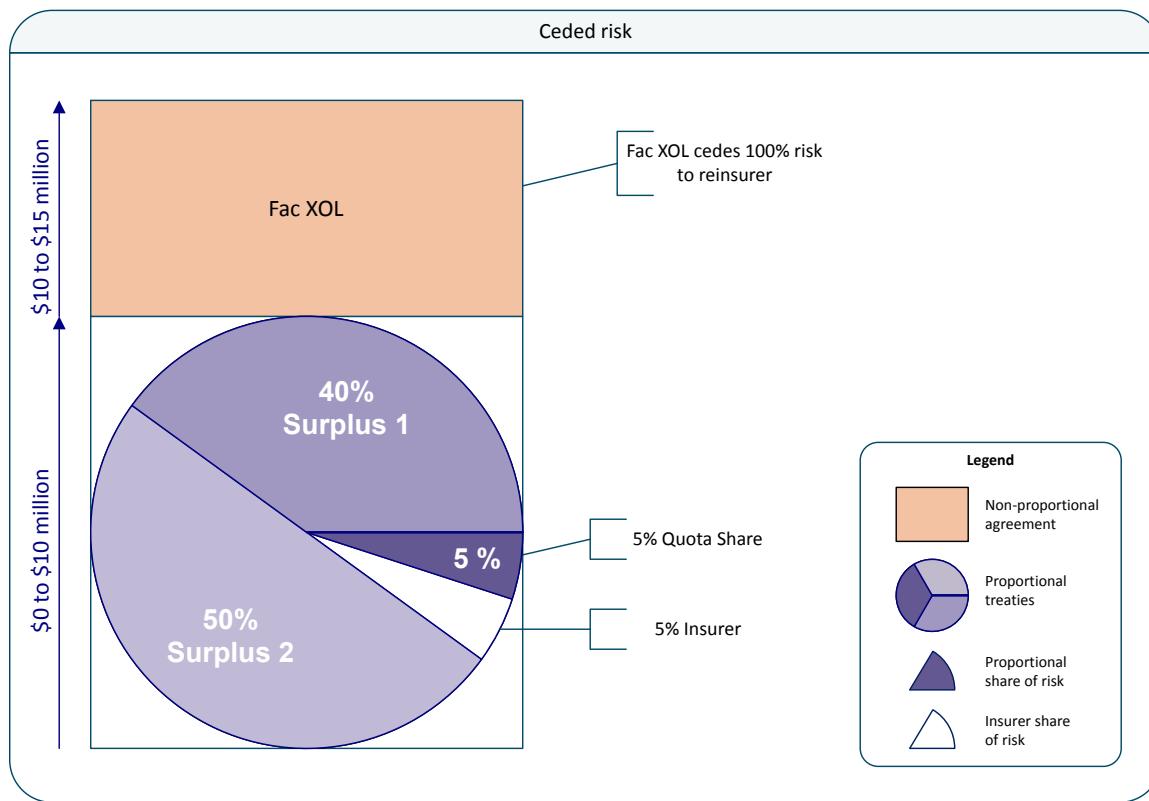
This example shows how PolicyCenter cedes premiums to several proportional agreements and to the facultative excess of loss agreement above the proportional agreements.

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
Proportional Treaties			
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%
Insurer's share	\$0 to \$1 million, insurer 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

Gross retention

The gross retention is the amount of risk retained by the insurer 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, an insurer 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.

See also

- “Modify the gross retention” on page 623

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 insurer determines the ceded premiums, commissions, and loss recoverables for the agreement. The insurer 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

- “Treaty or Facultative Agreement screen” on page 625

Differential rates of commission in reinsurance

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

- “Treaty or Facultative Agreement screen” on page 625

Location groups and reinsurance

Location groups provide a way to group locations that are located near each other. An insurer can use location groups to help assess their total risk for a given location.

An insurer 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, an insurer 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 insurer 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

- “Create a location group” on page 621
- “Geocoding locations” on page 327
- *Configuration Guide*

Reinsurance Management user interface

This topic describes the Reinsurance Management user interface in PolicyCenter.

Working with the Reinsurance tab

You can view and define reinsurance agreements and programs on the **Reinsurance** tab of a policy or policy transaction. This topic describes how to work with Reinsurance Management from the **Reinsurance** tab.

Search for agreements in reinsurance

About this task

You can search for all agreements, whether treaties or facultative agreements, and use search criteria to narrow the search.

Procedure

1. Select **Agreements** on the **Reinsurance** tab.

PolicyCenter displays the **Search Agreement** 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.

Search for all agreements in reinsurance

Procedure

1. Select **Agreements** from the **Reinsurance** tab to view the **Search Agreements** screen.
2. Click **Reset**.
3. In **Effective Date**, select **<none selected>**.
PolicyCenter displays all agreements.

Create a new treaty in reinsurance

Procedure

1. Select **Agreements** from the **Reinsurance** tab to view the **Search Agreements** screen.
2. Select **Actions→New Treaty** and select the treaty type.
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 “Treaty or Facultative Agreement screen” on page 625.
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 “Treaty or Facultative Agreement screen” on page 625.
6. Click **Update** to create the treaty.

Create a new program in reinsurance

About this task

A reinsurance program is a set of treaties put together by the insurer. In a multicurrency system, the program and its treaties must have the same currency.

Procedure

1. Select **Agreements** from the **Reinsurance** tab to view the **Search Agreements** screen.
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 “Reinsurance Program screen” on page 630.
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 “Reinsurance Program screen” on page 630.
6. Click **Update** to create the program.

Edit a program in reinsurance

About this task

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**.

Procedure

1. Access an existing program.

2. 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**

See also

- “Reinsurance Program screen” on page 630

Disable a program that has attached policies in reinsurance

About this task

You cannot delete a program that is attached to at least one policy. The **Delete** button is not accessible. Follow this process if you do not want this program to attach to any more policies.

Procedure

1. Edit the program.
2. Set the **Effective Date** and **Expiration Date** to the same value.

Create a new facultative agreement in reinsurance

Procedure

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 “Treaty or Facultative Agreement screen” on page 625.
5. Click **Update** to create the agreement.

Validate an agreement in reinsurance

About this task

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 runs prior to saving an already active agreement.

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:

Procedure

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**.

Make an agreement active in reinsurance

About this task

An agreement can have a status of **Active** or **Draft**.

Procedure

1. Click the **Reinsurance** tab.
2. 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 contains topics that describe how to work with the **Reinsurance** screen in a policy or policy transaction.

Add reinsurance to a policy

About this task

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 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.

Note: You can quote, bind, and issue a policy with projected reinsurance agreements.

Procedure

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, 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 Reinsurance Management links reinsurance to policies” on page 607.
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 633.
11. If you make a change, click **Save Draft** to validate all agreements on that policy term. **Save Draft** only appears in policy transactions.

Next steps

From the **Reinsurance** screen in a policy file, you can take additional actions as described in the following topics:

- “Create a location group” on page 621
- “View ceded premiums” on page 622
- “Gross retention” on page 613
- “Adding or linking to a facultative agreement” on page 623
- “Edit ceding parameters” on page 624

Create a location group

About this task

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.

Procedure

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. Note: 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 and reinsurance” on page 615
- “Geocoding locations” on page 327
- *Configuration Guide*

View ceded premiums

About this task

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.

This screen enables you to view ceded premiums in several different groupings: by policy transaction, by agreement, by agreement’s ceding across time.

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.

To view ceded premiums on a policy:

Procedure

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** – You can see how much premium was ceded at submission, as of the first policy change, or other transaction.
- **All agreements for a cost** – View each agreement's ceding in a calculation for each cost. You can see how premium was ceded for a single piece of premium for each job. For example, if \$100 was ceded in direct premium at submission, you can see how the premium was ceded to the applicable treaties.
- **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

See also

- “Calculating ceded premiums in Reinsurance Management” on page 610
- *Integration Guide*

Modify the gross retention

Procedure

1. In the policy transaction you created in “Add reinsurance to a policy” on page 620, 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 insurer 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		
							Lines	Start	Stop
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		
							Lines	Start	Stop
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.

See also

- “Facultative agreements” on page 599

Add a new facultative agreement

Procedure

1. Navigate to the **Reinsurance** screen in a policy or quoted policy transaction.
2. Under **Applicable Reinsurance**, select **Add Fac**→**Create New** and select one of the facultative agreement types. PolicyCenter displays the **Facultative** screen.
3. Enter the details of the agreement.
4. On the **Agreement Participants** tab, add at least one participant. A participant is a reinsurer participating in this agreement.
5. Click **OK** to create the agreement.

Link to an existing facultative agreement

Procedure

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**.

Edit ceding parameters

About this task

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.

Facultative agreements are added or removed on a per risk basis as described in “Adding or linking to a facultative agreement” on page 623. Therefore, facultative agreements do not have the **Edit** button.

To edit ceding parameters:

Procedure

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:

Value	Description
Included	Default. Include a risk in coverage for a treaty. This value does not appear in the Inclusion column.

Value	Description
Excluded	<p>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. Excluded appears in the Inclusion column.</p> <p>Excluding a risk from coverage has a particular behavior if a reinsurance program spans more than one term. If you exclude a risk from coverage by an agreement in a later term, PolicyCenter excludes the risk in that term and in later terms. PolicyCenter excludes the agreement by creating an inclusion row for the later terms. PolicyCenter does not exclude the agreement in prior terms.</p> <p>See AttachmentInclusion in the <i>Integration Guide</i>.</p>
Special Acceptance	<p>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. Special Acceptance appears in the Inclusion column.</p> <p>In the base implementation, this is the same as Included. You can customize this inclusion to meet your needs.</p>

Reinsurance Management screens

This contains topics that describe PolicyCenter screens related to Reinsurance Management.

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">• Per Even Treaty• Annual Aggregate Treaty• Quota Share Treaty• Surplus Treaty• Excess of Loss Treaty• Net Excess of Loss Treaty For facultative agreements, values are: <ul style="list-style-type: none">• Proportional Facultative Agreement• Excess of Loss Facultative Agreement• Net Excess of Loss Facultative Agreement
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.

Field	Description
	<p>For validation, the Expiration Date must be greater than the Effective date. 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.</p>
Status	<p>Data type: Typelist</p> <p>The status of the agreement. Values are:</p> <ul style="list-style-type: none"> • Draft – Not yet finalized or ready for use. • Active – Finalized and usable. • 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.
Currency	<p>In a multicurrency system, you can search by Currency. This drop-down list displays the currencies configured in the base application. For more information, see “Multicurrency features” on page 487.</p>
Coverage	
Ceded Share (%)	<p>Data type: XX.XXXX%</p> <p>The percentage ceded to the reinsurer. This field applies to the following types of agreements:</p> <ul style="list-style-type: none"> • Quota share treaties – Applies to the ceded share within the layer. No default value. • Facultative proportional – Applies to the ceded share for the agreement (total risk minus any risk covered by an excess of loss agreement). • 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%. <p>Note: This field does not apply to surplus treaties because they do not have a fixed value for ceded share.</p>
Coverage Limit	<p>Defines the upper bound on coverage.</p> <p>Note: Does not apply to proportional facultative agreements.</p>
Coverage Limit Indexed?	<p>Data type: Boolean</p> <p>Whether losses are subject to adjustment for inflation before being compared to the attachment point and limit of the agreement.</p> <p>Note: Applies to non-proportional agreements.</p>
Amount of RI	<p>Data type: Money</p> <p>Defines the maximum amount of reinsurance that can be recovered from the agreement.</p> <p>PolicyCenter calculates the value of this field from the attachment point and coverage limit by using a formula:</p> <ul style="list-style-type: none"> • For most types, Amount = (Coverage Limit - Attachment Point) * Ceded Share % • For quota share, Amount = Coverage Limit * Ceded Share % <p>Note: Does not apply to proportional facultative agreements. For proportional facultative agreements, the amount of reinsurance is equal to the Amount of Risk Ceded.</p>
Apply Only to the Gross Retention	<p>For net excess of loss treaties and facultative agreements only. If Yes, the amount ceded is limited to the layer below the gross retention that is not ceded to a proportional treaty.</p>
Amount of Risk Ceded	<p>Data type: Money</p> <p>Amount of risk ceded.</p> <p>Note: Applies to proportional facultative agreements.</p>
Attachment Point	<p>Data type: Money ≥ 0</p> <p>The lower limit for the start of coverage. For a Surplus Treaty, this is the lower limit assuming the maximum retention.</p> <p>Note: Does not apply to quota share or proportional facultative treaties.</p>
Attachment Point Indexed?	<p>Data type: Boolean</p> <p>Default: false</p> <p>Whether to adjust the loss costs by an inflation index before determining whether loss costs exceed the attachment point.</p> <p>Note: Applies to non-proportional agreements.</p>

Field	Description
Max Retention	<p>Data type: Money ≥ 0</p> <p>Default: Attachment Point</p> <p>The maximum underlying gross retention that the insurer can hold. This retention serves as the basis for the amount of reinsurance coverage provided.</p> <p>For validation, the maximum retention must be less than or equal to the attachment point.</p> <p>Note: Applies to surplus treaties.</p>
Lines	<p>Number of lines of coverage. PolicyCenter calculates this value by using the following formula: $(\text{Coverage Limit} - \text{Attachment Point}) / \text{Max Retention}$</p> <p>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 “Gross retention” on page 613.</p> <p>Note: Applies to surplus treaties.</p>
Start Line	<p>PolicyCenter calculates this value by using the following formula: $\text{Attachment Point} / \text{Max Retention}$</p> <p>Note: Applies to surplus treaties.</p>
Stop Line	<p>PolicyCenter calculates this value by using the following formula: $\text{Coverage Limit} / \text{Max Retention}$</p> <p>Note: Applies to surplus treaties.</p>
Premium and Commissions	
Min Deposit	Data type: Money
Premium	<p>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.</p> <p>Note: Applies to treaties.</p>
Deposit Payment Schedule	<p>Data type: Typelist</p> <p>The payment schedule for the deposit premium. Values are:</p> <ul style="list-style-type: none"> • Fully in advance • Quarterly in advance <p>PolicyCenter does not use this field in the default configuration.</p> <p>Note: Applies to treaties.</p>
Payable on Written	<p>Data type: Typelist</p> <p>When the ceded premiums and commission are payable. Values are:</p> <ul style="list-style-type: none"> • As Written – The ceded premiums and commissions are reported and considered payable as the written premium is recognized. • As Earned – The ceded premiums and commissions are considered payable as the underlying premium is earned. <p>PolicyCenter does not use this field in the default configuration.</p>
Commission (%)	<p>Data type: XX.XXXX%</p> <p>The commission that the insurer 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.</p> <p>Enter 0 if there is no commission. For example, many non-proportional agreements do not pay a commission.</p>
Set Differential Commission Rates	<p>Click this button to:</p> <ul style="list-style-type: none"> • Add a Commission % column for each participant in the Agreement Participants tab. • Remove the Commission (%) field. <p>For additional information, see “Shared reinsurance agreements” on page 614.</p>
Broker	<p>Data type: Contact</p> <p>The broker, if any, who helped arrange this agreement.</p>
Ceded Premium (flat amount)	Data type: Money

Field	Description
	<p>For facultative agreements, the flat amount that is paid to the reinsurer rather than ceding a proportional percentage.</p> <p>Note: Applies to non-proportional facultative agreements. Applies to proportional facultative agreements if the value is greater than 0.</p>
Set Differential Flat Premium	<p>Click this button to:</p> <ul style="list-style-type: none"> • Add a Flat Premium column for each participant in the Agreement Participants tab. • Remove the Ceded Premium (flat amount) field. <p>For additional information, see “Shared reinsurance agreements” on page 614.</p> <p>Note: Applies to facultative agreements.</p>
Ceding Rate (%)	<p>Data type: XX.XXXX%</p> <p>A percentage per dollar of the underlying net premium that is ceded to the reinsurer.</p> <p>For validation, the ceding rate must be greater than or equal to 0.</p> <p>Note: Applies to non-proportional treaties.</p>
Set Differential Ceding Rates	<p>Click this button to:</p> <ul style="list-style-type: none"> • Add a Ceding Rate (%) column for each participant in the Agreement Participants tab. • Remove the Ceded Rate (%) field. <p>For additional information, see “Shared reinsurance agreements” on page 614.</p> <p>Note: 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>Note: 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>Note: Applies to excess of loss facultative and net excess of loss facultative agreements.</p>
Other Terms	
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>Note: 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 ≥ 0</p> <p>Default: 0</p> <p>Notify the reinsurer if an individual large loss exceeds this threshold.</p> <p>Note: Applies to per risk agreements.</p>
Comments	<p>Data type: Text</p> <p>A text field for the insurer 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>

Field	Description
	<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"> • Gross Premium – Prior to netting out any other ceded premiums (including proportional cedings) • Net of proportional – Net premiums after ceding to proportional agreements but prior to any other non-proportional cedings • Net of per risk – Premiums after all cedings to per risk agreements but prior to any cedings to aggregate agreements • Net of all per event – Premiums after all cedings to per event agreements • Net of all prior – Premiums after all cedings <p>Note: Applies to all non-proportional agreements.</p>
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, an insurer 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>Note: 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"> • Loss Date Attachment (Earned Premium) – Ceded premiums are paid based on earned premiums that fall within the treaty period. • Loss Date Attachment (Written Premium) – Ceded premiums are paid based on written premiums that fall within the treaty period. • Policy Attachment – This value is available for excess of loss and net excess of loss treaties. <p>Note: 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 insurer's policy number.</p>
Ceding Rate (%)	<p>Data type: XX.XXXX%</p> <p>Appears when you click the Set Differential Ceding Rates button.</p> <p>If multiple participants share a non-proportional treaty, then they each negotiate a rate for their participation.</p> <p>Note: Applies to non-proportional treaties.</p>

Field	Description
Premium Share %	<p>Data type: XX.XXXX%</p> <p>Appears when you click the Set Differential Ceding Rates 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>Note: Applies to non-proportional treaties.</p>
Commission %	<p>Data type: XX.XXXX%</p> <p>Appears when you click the Set Differential Commission Rates 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 Select Differential Commission Rates button.</p> <p>How the participants share the commission.</p> <p>PolicyCenter calculates this field by using this formula:</p> $(\text{Risk Share} * \text{Commission Rate}) / \text{Agreement Overall Commission Rate}$ <p>The formula for the Agreement Overall Commission Rate is:</p> $\text{sum of } (\text{Risk Share} * \text{Commission Rate}) \text{ for each participant}$
Flat Premium	<p>Data type: Money</p> <p>Appears when you click the Set Differential Flat Premium button.</p> <p>Enter the premium amount that paid to each participant.</p> <p>Note: 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 599

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 Expiration Date must be greater than the Effective date .
Status	Data type: Typelist The status of the program. Values are: <ul style="list-style-type: none">• Draft – Not yet finalized or ready for use.• Active – Finalized and usable.• 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.
Currency	In a multicurrency system, the currency of the program. For more information, see “Multicurrency features” on page 487.
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 the <i>Configuration Guide</i> .
Single Risk Maximum	Data type: Money The maximum risk that the program covers. Increasing the Single Risk Maximum may require a corresponding increase in the Target Max Retention .
Test Retention	When you click the Test Retention button, PolicyCenter calculates the Implied Retention 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 Test Retention 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 for reinsurance

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"> • <none selected> • Coming Year – Find agreements with effective date starting at the current date through one year from this date. • Last Year – Find agreements in programs that are in effect on the current date and back one year from this date. • Custom Dates 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"> • <none selected> • Treaty • Facultative
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"> • <none selected> • Draft • Active
Currency	In a multicurrency system, specify the currency of the agreement. For more information, see “Multicurrency features” on page 487.

Search Programs screen for reinsurance

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"> • <none selected> • Draft • Active
Currency	In a multicurrency system, specify the currency of the program.
Effective Date	Values are: <ul style="list-style-type: none"> • <none selected> • Coming Year • Last Year • Custom Dates <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.
Reinsurable Risks	
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.
PML Amount	The probable maximum loss. Usually, PML equals TIV, but may be less if Probable Max Loss % is less than 100. You can use the PML Amount to adjust the amount of risk used for allocating reinsurance. Specify the PML Amount if it is extremely unlikely that there will be a loss as large as the TIV. 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.

Field	Description
Applicable Reinsurance	
Per Risk tab	
Total Risk	The total risk. This risk is the total insured value or PML, if adjusted by Probable Max Loss % .
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 Total Risk minus Covered by XOL .
Gross Retention	<p>The amount of risk retained by the insurer 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 Coverage Limit of the quota share treaty. If this risk does not have a quota share treaty, then the value defaults to the Max Retention of the first surplus treaty. Specify the Max Retention on the Reinsurance→Treaty screen.</p> <p>You can modify the gross retention. However, the value must be less than or equal to the Max Retention of the first surplus treaty. For an example, see “Gross retention” on page 613.</p>
Retained Prop Share (%)	The percentage of risk retained by the insurer of the total shared among proportional agreements.
Prop Retention	The amount of risk retained by the insurer. This amount is the Retained Prop Share (%) 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 insurer after ceding risk to all per risk reinsurance agreements.
Target Max Retention	<p>The target maximum retention. This retention is the value of Target Max Retention on the Reinsurance Program screen.</p> <p>If the insurer’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. For more information about the underwriting issue, see the <i>Configuration Guide</i>.</p>
Fac RI Needed	Facultative reinsurance needed. If the Target Max Retention 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 Target Max Retention .
Per Risk and Aggregate fields	
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 Ceded Share (%) on the Treaty or Facultative Agreement screen.
Max Ceding	Per risk only. The maximum amount of risk that can be ceded.
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 insurer 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 the <i>Integration Guide</i>.</p>
Prop %	Per risk only. For each proportional treaty, its percentage of the Shared Among Prop . This value is calculated according to the following formula: Ceded Risk / Shared Among Prop
Inclusion	Whether or not to include a risk in coverage by a treaty. This field has the following values: Included – Default. Include a risk in coverage for a treaty. This value does not appear in the Inclusion column.

Field	Description
Excluded	<p>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.</p> <p>Special Acceptance – Make a special inclusion that cedes risk to a treaty even though it would not normally. In the base implementation, this is the same as Included. You can customize this inclusion to meet your needs.</p> <p>Excluding a risk from coverage has a particular behavior if a reinsurance program spans more than one term. If you exclude a risk from coverage by an agreement in a later term, PolicyCenter excludes the risk in that term and in later terms. PolicyCenter excludes the agreement by creating an inclusion row for the later terms. PolicyCenter does not exclude the agreement in prior terms.</p> <p>See AttachmentInclusion in the <i>Integration Guide</i>.</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 Reinsurance Management selects a projected program if active program is unavailable” on page 608.

Part 8

Underwriting authority

Underwriting authority concepts

Underwriting is the process of examining, accepting, or rejecting insurance risks and classifying the ones that are accepted, in order to charge appropriate premiums for them. Underwriting authority in PolicyCenter provides an extensible underwriting infrastructure and a user interface for creating *underwriting rules*. In certain policy transactions, underwriting rules trigger the creation of *underwriting issues* on a policy transaction. Underwriting issues can block progress of a policy transaction. Underwriting *authority profiles* determine the types of underwriting issues that a particular underwriter can approve. When the underwriter approves an underwriting issue, the policy transaction can proceed.

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 insurer require that an underwriter approve any vehicle valued over \$100,000. When an agent adds a car valued at \$200,000 in a personal auto policy, an underwriting rules triggers creation of an underwriting issue. The underwriter must approve that issue before the agent can bind the policy.

See also

- *Configuration Guide*

Underwriting authority overview

Underwriting authority in PolicyCenter provides features for creating underwriting issues on a policy and for reviewing and approving those issues.

Underwriting rules specify the different kinds of underwriting-related issues. The rule specifies when to raise that issue on a policy. Underwriting rules also specify the point at which policy transactions are blocked by unresolved issues. One or more *authority profiles* are assigned to individual users. These profiles allow the user to approve blocked issues within the levels specified in the *authority grants* for each specific issue.

The base configuration contains authority profiles for agents, underwriters, and an underwriting manager. You can modify the authority profiles and create new ones. 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 or quote release, the agent cannot view the quote. The quote is not visible until the underwriter approves the issue and releases the policy back to the agent. The agent can then 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. This is accomplished either by defining the rule condition in **Administration→Business Settings→Business Rules**, or by writing Gosu code to raise the issue. 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.

Underwriting authority components

Underwriting authority contains the following major components:

- **Underwriting Rule** – Includes the name of the rule and describes how the policy transaction is affected by underwriting, and how issues can be approved. The rules includes the condition under which it will raise an underwriting issue. PolicyCenter provides a user interface in **Administration→Business Settings→Business Rules** for defining underwriting rules. Alternatively, you can implement the underwriting rule in Gosu code instead rather than through the business rules interface.
- **Underwriting Issue** – A specific occurrence on a policy transaction where an underwriting condition was true.
- **Authority** – The ability to approve underwriting issues created by a rule, and to what degree.
- **Authority Profile** – A group of authorities that can be given to a user, allowing the user to approve underwriting issues that were created by a particular rule.

Choosing between validation and underwriting authority

Both validation and underwriting authority provide mechanisms to stop the progress of a policy transaction. In the base configuration, validation is checked before underwriting authority.

Use validation to prevent the progress of policy transactions that nobody can sign off on. Validation checks for:

- Missing or inconsistent data – Including things like a vehicle on a personal auto policy without a VIN at bind time, as well as conflicting coverages or terms.
- Structurally inappropriate policies – For example, some insurers do not ever permit a personal auto policy with no vehicles.

Use underwriting authority to restrict who can progress a policy transaction that at least somebody in the organization could sign off on. If only certain people within the insurer organization can approve writing a policy in California, then use an underwriting rule.

However, if the insurer will never write a policy in the state of California under any circumstances, then use a validation rule.

Underwriting issue flow

Underwriting rules determine how and when to create underwriting issues. The underwriting rule specifies when to check the policy for particular conditions; this is the *checking set* of the rule. The underwriting rule specifies whether and where to block progress of the policy transaction; this is the *blocking point*.

In the following example, you are Alice Applegate, an underwriter. In PolicyCenter, you start a policy transaction for a personal auto policy for Ray Newton. You populate the policy by entering drivers and vehicles, and selecting coverages. The new vehicle cost is \$110,000.

At specific points during the submission PolicyCenter checks the rules to evaluate whether to raise underwriting issues. These specific points, such as before quote or bind, are defined in the checking set. If rule criteria is met, then PolicyCenter raises an underwriting issue of that type. PolicyCenter has a High Value Vehicle underwriting rule that creates an underwriting issue if new vehicle cost is greater than \$100,000. The rule checks for the issue before quote and blocks bind if the issue is not approved.

When you click **Quote**, PolicyCenter evaluates the policy data, and raises a High Value Vehicle underwriting issue. PolicyCenter also displays a message the underwriting approval is required before binding the policy. You can view

this issue on the **Risk Analysis** screen. If you click **Bind**, PolicyCenter displays this issue on the **Issues that block Binding** screen.

To approve an issue, your authority profile must contain sufficient authority. If you approve this issue, the policy can be bound.

At each checking set, PolicyCenter checks for *orphaned issues* and removes them. Orphaned issues are underwriting issues that were generated by this checking set but are no longer an issue. For example, if you change the new vehicle cost to \$90,000, the High Value Vehicle issue becomes an orphan and is removed at that checking set.

At each blocking point, PolicyCenter checks for issues blocking at the current blocking point and any earlier blocking points.

See also

- “Authority profiles” on page 721

Implement underwriting authority

About this task

Implementing underwriting authority requires changing various areas of PolicyCenter. This topic provides steps to help guide you through the process.

IMPORTANT Once an underwriting rule has been deployed to production, do not change it without fully understanding how those changes will impact your system. PolicyCenter prohibits certain types of changes, such as changing the value comparator. Changing the rule condition may make it impossible to compare system metrics for the rule before and after the change is deployed. Similarly, changing the issue key may cause more or fewer underwriting issues from this rule on individual transactions. Consider only making change to rules that have been deployed to production when the rule is incorrect. In most cases, Guidewire recommends that you create new underwriting rules when changing the condition or other values that affect rule behavior.

To implement underwriting authority:

Procedure

1. Define the underwriting rules for your implementation.
 - a. Review the existing underwriting rules in the base configuration to determine if they need to be modified or removed.
 - b. Determine whether you need to add new underwriting rules.For more information, see “Guidelines for designing underwriting rules” on page 645.
2. In PolicyCenter, create authority profiles in the **Administration→Users & Security→Authority Profiles** screen. Each authority profile grants authority for specific issues. For more information, see “Authority profiles” on page 721.
3. In PolicyCenter, add the authority profiles to individual users in the **UW Authority** tab in the **User** screen. For more information, see “Working with authority profiles” on page 722.
4. In PolicyCenter, grant users permissions as appropriate for their role.

The permissions that apply to underwriting issues are:

- **View risk analysis evaluation issues** – The code for this value is `viewriskevalissues`.
- **View risk analysis referral reasons** – The code for this value is `viewriskrefreasons`.
- **Edit Lock Override** – The code for this value is `editlockoverride`.
- **Quote Hide Override** – The code for this value is `quotehideoverride`.
- **Reject UW Issues** – The code for this value is `uwreject`.
- **Reopen UW Issues** – The code for this value is `uwreopen`.

- **Approve all UW Issues** – This permission grants the ability to approve any issue to any level. It is intended only for resolving missing authority issues in a time-critical situation. The code for this value is `uwapproveall`.

For more information about permissions, see “Security: roles, permissions, and the community model” on page 699.

5. Specify the PolicyCenter user that will process automated renewals.

Note: Automated processes, such as automated renewals, treat all underwriting issues as auto-approvable. For more information, see “Approvals of underwriting issues” on page 653.

- In your implementation of the Renewal plugin, implement the `getAutomatedRenewalUser` method to return this user. In the default configuration, this method returns the user `renewal_daemon`. (Be sure to assign appropriate authority profiles to the `renewal_daemon` user. Guidewire suggests that the authority profiles be similar to other users.)

For details about the renewal plugin, see the *Integration Guide*.

- Add that user in PolicyCenter if necessary. Assign authority profiles to that user. See “Assign an authority profile to a user” on page 723.

6. If you will be automatically processing policy changes, specify the user for PolicyCenter to use. You can automatically start policy changes by using the `startAutomaticPolicyChange` method in the `PolicyChangeAPI`. In the default configuration, this method uses the `policychange_daemon` user. For details about this API, see the *Integration Guide*.

7. You can also make the following modifications.

- Add a new checking set and change the job processes to check for it. See the *Configuration Guide*.
- Add a new value comparator to use with issues. See the *Configuration Guide*.
- Add a new value formatter to use with issues. See the *Configuration Guide*.

Underwriting rules

Underwriting rules are part of underwriting authority in PolicyCenter. Underwriting authority provides an extensible infrastructure and a user interface for defining underwriting rules. Underwriting rules specify when to create underwriting issues on policy transactions.

Overview of underwriting rules

If a policy contains items that may require underwriter review, underwriting rules trigger the creation of underwriting issues in policy transactions. Underwriting rules specify when to check and under what conditions to raise underwriting issues in the policy transaction. PolicyCenter provides a user interface for creating underwriting rules.

Some of the settings that you can specify in an underwriting rule are:

- Which policy transactions, policy lines, and jurisdictions check for this underwriting rule.
- What conditions must exist on the policy transaction for the underwriting issue to be created.
- At which points in the policy transaction to check the policy for these conditions.
- Whether to prevent progress of the policy transaction until the underwriting issue has been approved.

For example, when an agent creates a new policy, certain terms of that policy may need underwriting review before the policy can be bound. An underwriting rule requires that an underwriter approve any vehicle valued over \$100,000. When an agent adds a car valued at \$200,000 in a personal auto policy, an underwriting rule triggers creation of an underwriting issue. The underwriter must approve that issue before the agent can bind the policy.

Underwriting issues are always raised when the condition is true, even if the current user has the authority to approve the issue. In the underwriting rule, you can define issues to be automatically approved if an underwriter has the authority to approve an issue on a policy transaction they are working on.

See also

- *System Administration Guide*

Business rule execution flow

In general, PolicyCenter business rules have the following execution flow:

- PolicyCenter application logic triggers the evaluation of business rules at certain points in the application flow, based on rule implementation. Only those rules that match a triggering action and a triggering entity are candidates for evaluation.
- Business rule logic uses applicability criteria, defined in the **Applies To** area of the Business Rules editor, to filter rules for evaluation based on the current state of the data. For example, if **Jurisdictions** = California, PolicyCenter evaluates only those rules with a jurisdiction of California.
- After PolicyCenter determines the list of rules eligible for execution, it evaluates the rule conditions for each rule. PolicyCenter then creates the set of rules for which the rule conditions evaluate to true and executes the actions for this set of rules.
- Rule logic caches the rules for execution at every triggering point. PolicyCenter flushes this rule cache at every rule edit, rule import, and application restart.

Accessing business rules

You access and manage business rules in the **Administration** tab using the **Business Settings**→**Business Rules** menu link.

The **Underwriting Rules** screen displays a list of existing underwriting rules, if any.

Business rules roles and permissions

You need appropriate permissions and roles to access the **Business Rules** screens and manage rules. A set of permissions is available to enable users to create, edit, deploy, and manage business rules.

The Rules Admin role can be assigned to users for business rules.

See also

- *System Administration Guide*

Setting up business rules

In the base configuration, PolicyCenter provides a set of rules that can be imported into a development environment. Once imported, you can review and edit the set of rules using the **Business Rules** screens.

When the set of rules has been edited, reviewed and approved, it can be exported and re-imported into a production environment and deployed.

There are a number of configuration parameters for enabling and deploying business rules.

See also

- *Configuration Guide*

Business rule states

At initial creation, all business rules are in the Draft state. As you begin the process of reviewing, evaluating, and updating a business rules, its state can vary and progress.

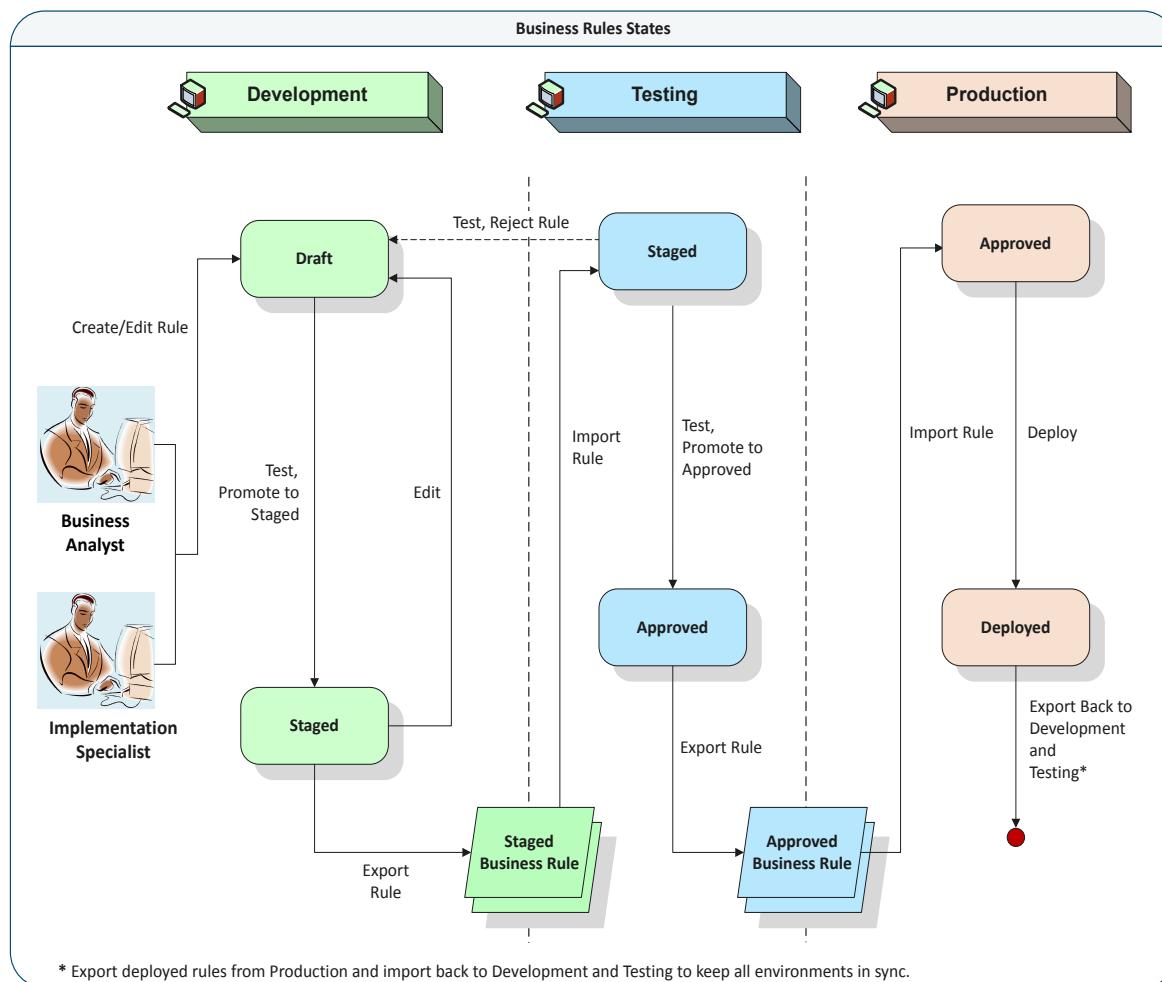
The Guidewire PolicyCenter business rules have the following sequence of states.

Draft	PolicyCenter assigns a status of Draft after you save the initial version of the rule. A rule reverts to Draft status whenever the rule undergoes any type of editing. It is not possible export a rule in the Draft state.
Staged	You manually move a rule in the Draft state to the Staged state, after you complete rule editing. Typically, this is the point in the rule lifecycle that you export the rule from the development environment and import the rule into the testing environment.

Approved You manually move a rule from the Staged state to the Approved state, usually in the testing environment after you complete the rule evaluation phase. Typically, this is the point in the rule lifecycle that you export the rule from the testing environment and import it into the development environment.

Deployed You manually move a rule from the Approved state to the Deployed state. Typically, this occurs after you import the rule into the production environment.

The following diagram describes the various states associated with a business rule, as it is created, edited, and deployed in development and production environments.



See also

- [System Administration Guide](#)

Guidelines for designing underwriting rules

You design underwriting rules in the **Administration**→**Business Settings**→**Business Rules**→**Underwriting Rules** screen. PolicyCenter uses these underwriting rules to create underwriting issues on policies. In addition, you can use Gosu code to implement underwriting rules. For example, you can use Gosu code for underwriting rules with rule conditions too complex to define in the **Rule Condition** editor.

IMPORTANT You must create and manage the characteristics of the set of underwriting rules very carefully. Guidewire recommends that you do not remove underwriting rules once they have been deployed on a production system. In addition, do not change the **Code**, **Value Comparator**, **Blocking Point**, **Checking Point**, and **AutoApprovable** fields for underwriting rules after the rule is in production. It is permissible to alter the fields relating to approval defaults, along with the name or description of the issue. You must also maintain authority grants that reference these older underwriting rules. However, you can create new underwriting rules to use for future occurrences of this issue.

When defining underwriting rules for your implementation, determine whether to modify or remove existing underwriting rules, and whether to add new underwriting rules.

To define underwriting rules, consider the following questions.

1. **What is the purpose of this underwriting rule?** Does the rule raise an underwriting issue when the driver is more than 80 years old? Or does the rule raise an issue when the business prepares fried foods?
2. **How do you detect when the underwriting rule creates an underwriting issue?** For example, which fields on the policy period trigger creation of this underwriting issue? What logic determines triggers the creation?
3. **Must you raise an issue based on conditions outside of a policy transaction and policy data?** If so, you may need to create an underwriting referral reason. The underwriting referral reason triggers creation of an underwriting issue.
See “Underwriting referral reasons raise underwriting issues” on page 652.
4. **What type of value is associated with this underwriting issue?** Does the value affect which underwriters can approve that issue?

A value can be a number, a currency amount, a jurisdiction or set of jurisdictions, a class code or a range of class codes. It can also be another type of value that can be easily represented as a string.

An underwriting issue requires a value if that value affects which underwriters can approve it. For example, suppose you have an underwriting rule *Driver is under 25 years old*. If the same set of underwriters can approve the issue whether the driver is 17 or 24, then the rule does not need a value. However, suppose some underwriters can approve the issue if the driver is at least 21, but others can approve the issue when the driver is only 16. In this case, use the age as the value associated with the underwriting issue.

You can set values for underwriting rules even when not required. For example, if age is not strictly necessary, there is no harm in associating it with the underwriting rule. In the default configuration, the underwriter sees the value when PolicyCenter displays the underwriting issue.

See “Underwriting issues with values” on page 647.

5. **If there is a value associated with this underwriting issue, can you use an existing value comparator?**

In the base configuration, PolicyCenter provides value comparators for numeric values and for jurisdictions. You can add additional comparators if necessary. Adding a new value comparator is complex. Therefore, consider carefully before adding a new one.

See the *Configuration Guide*, and the *Configuration Guide*.

6. **If there is a value associated with this underwriting rule, can you use one of the existing value formatters?**

In the default configuration, PolicyCenter provides value formatters for integer, currency, jurisdictions, sets of jurisdictions, and string values. If these formatters are not sufficient, you may need to add a new value formatter. Although not as complex as adding a new comparator, consider carefully before adding a new one.

See “Business rule: value formatter type” on page 648 *Configuration Guide*.

7. **Which policy transaction types will check for this underwriting issue?**

The checking set for the underwriting rule determines when PolicyCenter checks for the issue. Determine the policy transaction types (such as submission, renewal, issuance) that will check for and raise this underwriting issue. Within each policy transaction, determine which transition points (such as quoting or binding) check the underwriting rule. Does the policy transaction check this underwriting rule at one point or multiple points?

In the base configuration, underwriting issues are available in submission, issuance, policy change, reinstatement, rewrite (includes rewrite new account), and renewal policy transactions. Through configuration,

you can extend underwriting to other policy transaction types, such as cancellations. For more information about which policy transactions provide underwriting in the base configuration, see “Key features of policy transactions” on page 85.

Underwriting issues that will not change after quoting may be checked for only at quoting. There is no reason to check the underwriting rule again unless the policy needs re-quoting.

Other underwriting issues require multiple checking sets. For example, suppose PolicyCenter receives payment delinquency information from a billing system. You can raise the underwriting issue at the next transition point (quoting, binding, issuance) no matter when it is checked for.

Note: In general, Guidewire recommends that you raise underwriting issues as early in the policy transaction as possible.

See “Checking sets and blocking points for underwriting issues” on page 649, the *Configuration Guide*.

8. Does this underwriting rule block progress of the policy transaction?

Consider the point at which the underwriting issue blocks progress. For example, a rule checks and raises fire hazard issues before quoting a policy. If these issues are not approved, the policy transaction cannot be bound. The issue has a blocking point at bind

See “Checking sets and blocking points for underwriting issues” on page 649 and the *Configuration Guide*.

9. Is the underwriting rule too complex to be implemented in the Underwriting Rules user interface?

- If just the rule condition is too complex, create a custom utility function to implement the rule condition. In the **Underwriting Rules** user interface, have the **Rule Condition** call this utility function.
- If the underwriting rule has outcomes that are more complex than just adding an underwriting issue, use a Gosu implementation.

See the *Configuration Guide*.

See also

- “Underwriting rules” on page 643
- *Configuration Guide*

Underwriting issues with values

An underwriting rule can create *value issues* which have a value associated with them. 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.

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.

Business rule: value comparator values

The **Value Comparator** on the **Advanced Info** tab of a business rule specifies how to compare the issue value. You can specify:

- If the value of an issue is within the authority granted to a user
- The limits of what an underwriter can approve

The **Comparison** column of an **Authority Profile** in **Administration→Users & Security** specifies whether the value of an approval is within the authority granted to a user.

In the default configuration, the values are:

Value	Code	Description
At least	Numeric_GE	The issue value must be greater than or equal to the approval or authority grant value. Select this value if the issue has an associated value where a smaller number is associated with more risk, such as a deductible. Treat the value as a <code>BigDecimal</code> .
At least (monetary)	Monetary_GE	The issue value must be greater than or equal to the approval or authority grant value. Select this value if the issue has an associated value where a smaller number is associated with more risk, such as a deductible. Treat the value as a <code>MonetaryAmount</code> .
At most	Numeric_LE	The issue value must be less than or equal to the approval or authority grant value. Select this value if the issue has an associated value where a larger number is associated with more risk, such as total premiums or total insured value. Treat the value as a <code>MonetaryAmount</code> .
At most (monetary)	Monetary_LE	The issue value must be less than or equal to the approval or authority grant value. Use this comparator for currency amounts. Select this value if the issue has an associated value where a larger number is associated with more risk, such as total premiums or total insured value. Treat the value as a <code>MonetaryAmount</code> .
In set	State_Set	Treat the authority grant or approval value as a set of jurisdictions, and the issue value must be within that set.
None	None	Has no associated value. Use for issues that either exist or do not exist.

The value comparator corresponds to the `UWIssueType.Comparator` property.

See also

- **Value Comparator** field in “Advanced Info tab on Underwriting Rule screen” on page 659
- *Configuration Guide*

Business rule: default value assignment type

The **Default Value Assignment Type** on the **Advanced Info** tab of a business rule specifies how to compute a default approval value from the value of the issue. PolicyCenter uses this value if **Value Comparator** is **At least** or **At most**.

Values are:

Value	Description
<none>	For issues that have a non-numeric value or that do not have an associated value.
Fixed	The default approval value is copied directly from the issue value.
Offset Amount	The Default Value Offset Amount is added to (or subtracted from, depending upon the implementation of the Value Comparator) the issue value to produce the default approval value. Use only with issues that have a numeric value.
Offset Percentage	The Default Value Offset Amount is treated as a percentage increase or decrease, and the issue value is offset by that percentage to produce the default approval value. Use only with issues that have a numeric value.

The default value assignment type corresponds to the `UWIssueType.DefaultValueAssignmentType` property. The `UwValueAssignmentType` typelist defines these values.

Business rule: value formatter type

The **Value Formatter Type** on the **Advanced Info** tab of a business rule specifies how to display the value in the underwriting issue.

In the base configuration, the values are:

Value	In UI	Description
Age	•	Formats the value as an age followed by the abbreviation yrs.
Currency		Formats the value as the currency associated with the current locale.
Integer	•	Formats the value as an integer.
MonetaryAmount		Formats the value as a MonetaryAmount, which contains a value and a currency.
Number	•	Formats the value as a BigDecimal.
StateSet	•	Formats the value as a jurisdiction.
USD		Formats the value as a U.S. dollar (with cents).
USDBrief		Formats the value as a U.S. dollar (without cents).
Unformatted	•	Does not format the value.
Units	•	Formats the value as an integer followed by the word units.
TestFormatter		Used only for testing.

The In UI column indicates values that appear in Rule screen of the base configuration. If you add hidden values such as Currency, PolicyCenter supports those values without further configuration.

The value formatter type corresponds to the `UWIssueType.ValueFormatterType` property.

See also

- **Value Formatter Type** field in “Advanced Info tab on Underwriting Rule screen” on page 659
- *Configuration Guide*

Checking sets and blocking points for underwriting issues

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 underwriting rule. 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 underwriting rule. For example, PolicyCenter prevents a user from binding a policy until high-value vehicle issues are approved.

Checking set values

In an underwriting rule, **Checking Set** specifies when to check for existence or absence of an underwriting rule. The evaluator classes further enforce which jobs check for that issue type. In some cases, PolicyCenter uses this value to filter issue types for display in the user interface.

The checking set represents the place in a policy transaction that a rule is checked.

The following table describes the checking set values in the base configuration.

Value	In UI? UI?	Description
All		PolicyCenter checks this rule at all blocking points. PolicyCenter removes an existing underwriting issue if the evaluators do not trigger this type of issue. Issues created by this checking set may come from external systems in a way that is not necessarily synchronized with the job flow. For example, a claim system sends notice of excessive claims, or a billing system sends notice of a delinquent account. If you have an issue that cannot change once the policy is quoted (without re-editing the policy), do not use this checking set.
Manual	•	The underwriting issue is created when the user manually added an issue to the policy period. PolicyCenter displays Manual issues on the Risk Analysis screen.
MVR		For motor vehicle record underwriting issues. PolicyCenter checks for this issue at quote, quote release, bind, and issuance.
PolicyRenewalAPI	•	PolicyCenter checks this rule prior to calling policy renewal web services. Policy renewal web services provides methods for managing renewals in PolicyCenter. See the <i>Integration Guide</i> .
PreBind	•	For underwriting issues that can only be detected immediately prior to binding the job. PolicyCenter checks this rule prior to binding.
PreIssuance	•	For underwriting issues that can be detected immediately prior to issuance. For submission policy transactions, PolicyCenter checks this rule prior to issuing the policy. If the policy transaction does not have an issuance step, PolicyCenter checks this rule prior to binding.
PreQuote	•	For underwriting issues that can be detected on the policy period even before quote, such as issues related to data entered in draft mode. PolicyCenter checks this rule prior to generating a quote.
PreRateRelease		PolicyCenter checks these rule underwriting rules after quoting and before the rate or quote is released.
PreQuoteRelease	•	For underwriting issues that need information from a valid quote or that can be detected after a valid quote. For example, use this checking set for issues that depend on the premium. PolicyCenter checks this rule after a valid quote, but before releasing the quote.
Question		For underwriting issues is based on answers to a question set. PolicyCenter checks this rule before quote. For information on how to associate underwriting rules with questions, see the <i>Product Model Guide</i> .
Referral		For underwriting issues is related to underwriting referral reasons on the policy. PolicyCenter checks this rule at all blocking points. PolicyCenter displays these issues in the UW Referral Reasons tab of the Risk Analysis screen.
		You can also add a referral reason to a policy by using the <code>addReferralReason</code> API.
RegulatoryHold		For underwriting issues related to policy holds with the Regulatory Hold hold type. PolicyCenter checks this rule at all blocking points. The user is notified each time the job advances to the next step.
Reinsurance		For underwriting issues related to reinsurance. For example, you can use this checking set to raise an issue if a policy has insufficient reinsurance. PolicyCenter checks this rule at all blocking points except quote.
Renewal	•	For underwriting issues related to renewals. PolicyCenter checks this rule in renewal policy transactions at quote and issuance.
Upgrade		In certain cases, Guidewire uses this value when upgrading PolicyCenter. Do not use this value for any other purpose.
UWHold		Use for underwriting issues related to policy holds with the Underwriting Hold hold type. PolicyCenter checks this rule at all blocking points. Therefore, the user is notified each time the policy transaction advances to the next step.

The In UI column indicates the values that appear in the Rule screen of the base configuration. The `TriggeringPointKey.TF_UWRULECHECKINGSETFILTER` filter narrows the choices in the user interface. Modify this filter to add hidden values such a **RegulatoryHold**.

Except for **All**, a rule can be associated with only one checking set.

Checking set values correspond to the `UWIssueType.CheckingSet` property. The `UWIssueCheckingSet` typelist defines these values.

See also

- *Configuration Guide*

Blocking point values

In an underwriting rule, **Blocking Point** specifies the point at which this issue blocks progress of a policy transaction. The blocking point values, listed in priority order from lowest to highest, are:

Value	Code	Description
Non-Blocking	<code>NonBlocking</code>	Sometimes referred to as informational issues. These issues do not block progress unless specifically rejected by a user.
Blocks Issuance	<code>BlocksIssuance</code>	Prevents issuance for submission policy transactions. Prevents binding for policy transactions that do not have a separate issuance step.
Blocks Bind	<code>BlocksBind</code>	The issue blocks binding the policy.
Blocks Quote Release	<code>BlocksQuoteRelease</code>	Allows quoting to be performed, but unauthorized users cannot view the quote until an underwriter has the opportunity to review and approve it. The job cannot progress further.
Blocks Rate Release	<code>BlocksRateRelease</code>	Blocks releasing a policy period rate. If two-step quoting is enabled in the policy line, this prevents the user from accessing the Quote screen. For example, although the policy is rated, you may not want the user to access the Quote screen until the policy has underwriting approval. In one-step quoting, this blocking point blocks quote release. The <code>BlocksRateRelease</code> blocking point is treated slightly differently from other blocking points. At the <code>BlocksRateRelease</code> blocking point, PolicyCenter only checks the <code>PreRateRelease</code> checking set, and does not check the <code>All</code> checking set. This is done for performance reasons, to make the first step of two step quoting execute quickly. If necessary, you can modify this behavior by making a change in the <code>JobProcessUWIssueEvaluator</code> class.
Blocks Quote	<code>BlocksQuote</code>	Prevents quoting and steps leading up to quoting. Use for issues that require intervention before the rating engine is called. You can use this blocking point for other types of issues, as well. However, if a policy transaction has a Blocks Quote issue, PolicyCenter will not call the rating engine, and the policy transaction remains in draft state. This blocking point also blocks rating, which is the first step in two-step quoting. Do not use for issues that need information from a valid quote. For example, do not use for issues related to total premium.
Rejected	<code>Rejected</code>	The most restrictive value. Rejects the issue. A rejected issue prevents the policy transaction from crossing any blocking point.

The blocking point corresponds to the `UWIssueType.BlockingPoint` property.

Note: Consider using the **Blocks Quote Release** blocking point rather than **Blocks Quote**. Some data, such as quote modifiers, are generated as a result of obtaining the quote. If this data triggers a **Blocks Quote** issue, the user may have difficulty resolving the issue. The difficulty is because the user interface, as a result of not permitting quote, may no longer allow the user to edit those fields.

Blocking points and checking sets are related, but not the same. In general, there is a checking set for each blocking point. However, there are other special checking sets as well as special blocking points. Issues need not have the

same checking set and blocking point. For example, an issue can be created pre-quote, but not block until quote release or binding. The special blocking points are:

- **Non-Blocking** – An issue does not block progress.
- **Rejected** – A rejected issue blocks further progress because it prevents the job from crossing any blocking point.

In addition, there are blocking points for quote, quote release, bind, and issuance.

The blocking point values have a priority order. When the job is at a blocking point, the job progress is blocked by any open issue that blocks at that priority or at a higher priority. Generally, higher priority blocking points occur earlier in the job. For example, a job has been quoted and is waiting to be bound. A loss-control specialist adds a referral reason to the policy that blocks quote. The resulting issue blocks binding the policy, even though the **Blocks Quote** blocking point has already been passed. The user can approve the issue then bind the policy without needing to obtain a new quote. In the default configuration, raising a lower priority blocking issue does not invalidate already completed steps.

Underwriting referral reasons raise underwriting issues

Underwriting referral reasons cause underwriting issues to be raised the next time a policy transaction is run on for that policy. Referral reasons raise underwriting issue types that have the **Checking Set** equal to **Referral**. In the base configuration, the Underwriting Rule user interface does not display **Referral**.

Use underwriting referral reasons 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 use referral reasons as a convenient way to add fully-formed underwriting issues to a policy from an external application. These fully-formed issues do not require additional data from the policy in PolicyCenter. Using them in this way requires that the external application has knowledge of the underwriting guidelines, such as when to consider claim levels as excessive.

Underwriting referral reasons can be added by a PolicyCenter user. For example, a PolicyCenter user adds a referral reason in response to an on-site visit or discussion with the insured.

Underwriting referral reasons can be added by an external system 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 a underwriting referral reason on the policy. The next time a policy transaction runs on the policy, PolicyCenter creates an underwriting issue for this referral reason.

A user cannot approve an underwriting referral reason, but the user can remove it. For example, a visit to a customer site reveals numerous safety violations, so you add an underwriting referral reason to the policy. At a subsequent visit, you find that the safety violations have been fixed. Therefore, you remove the underwriting referral reason from the policy.

In some situations it may be more appropriate to avoid referral reasons, and use underwriting issues directly. You can configure PolicyCenter to make calls to an external application, and extend the **Policy** entity to store data from the external application. For example, a claim system does not know the total premium of the policy and cannot compute a loss ratio. So it cannot send a fully-formed referral reason to PolicyCenter. PolicyCenter can retrieve the loss information from the claim system when it evaluates the appropriate checking sets. (The default integration with a claim system implements this retrieval. See “Accessing summary loss information from the claim system” on page 784.) Alternately, the external claim system could use an API to send loss information to PolicyCenter. PolicyCenter could use this information when raising underwriting issues.

See also

- “Add underwriting referral reasons” on page 674
- *Configuration Guide*

Approvals of underwriting issues

Once an underwriting issue is raised, the issue can be approved in the following ways:

- User manually approves the issue
- Issue is approved automatically depending upon user's authority profile
- Issue is approved automatically during automated renewal and policy change transactions

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.

Issues defined as auto-approvable are given automated approvals during issue evaluation, provided that all of the following are true:

- The issue does not already have an approval or rejection.
- The approval will unblock progress.
- No non-auto-approvable issues block progress.
- No other issues block progress.
- The user has the authority to approve all blocking, auto-approvable issues to their default values.

If the preceding conditions are true, then PolicyCenter automatically, and transparently, makes that set of approvals as the user progresses the policy transaction.

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 underwriting rule. Auto-approvable issues are not automatically approved if there is a manual approval on the issue.

Note: During automated processing, such as unattended renewals, the auto-approvable approval logic applies to all issues, including those that are not identified as auto-approvable.

In an underwriting rule, use the **Auto-approvable** field on the **Advanced Info** screen to specify whether this issue is treated as a auto-approvable issue. Auto-approvable issues can be automatically approved if the user has an authority grant that permits an approval at the default level. If PolicyCenter can approve all blocking auto-approvable issues at the default level, then PolicyCenter approves all blocking issues. Otherwise, PolicyCenter approves none. The default level is the amount in the authority grant plus any **Default Value Offset Amount** from the underwriting rule.

Set **Auto-approvable** to **No** if you want an underwriter to explicitly review and approve the issue, even if they have the authority to approve it without a referral. Set **Auto-approvable** to **Yes** if you want an underwriter with sufficient authority to not explicitly be required to approve the issue. Auto-approvable has no affect on approvals when the current user lacks the authority to approve the issue.

Set **Auto-approvable** to **Yes** for issues that, while they prevent unauthorized users from progressing the job, they are not unusual if the user is authorized. Automatically approving common issues allows the user to focus on the more important and unusual issues. For example, you may want to automatically approve issues raised to confirm the jurisdictions in which an agent is licensed or authorized to write business. Thresholds on sizes or types of policies or changes might also be automatically approved.

IMPORTANT Guidewire recommends that you define all auto-approvable issues with **DefaultValueAssignmentType** set to **Fixed**. This setting avoids a situation where the user can approve the issue, but the default approval is outside of their authority.

An automated approval has an **AutomaticApprovalCause** which you can use to identify these approvals in the user interface. In the default configuration, this value is *jobNumber@BlockingPoint* for automated approvals that happen during normal job processes.

Auto-approvable corresponds to the **UWIssueType.AutoApprovable** property.

See also

- *Configuration Guide*

Automated renewals and policy changes

In automated renewals and policy changes, all issues are treated as auto-approvable issues.

Approval duration values

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.

If any change to the policy transaction requires that the approval for this issue be removed, set **Allow Edit** to **No**. If the approval remains valid even if the user makes other changes to the transaction, set **Allow Edit** to **Yes**.

Use **Approval Duration** to 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 underwriting rule, you can configure the default duration. While approving the issue, the underwriter can change the duration.

In an underwriting rule, the **Duration** specifies the default duration on new approvals. Use this property to indicate when an approval will expire.

Values are:

Value	Description
Next Change	The approval expires on the next issuance, policy change, renewal, or rewrite policy transaction.
End of Term	The approval expires on the next renewal or rewrite policy transaction.
One Year	The approval expire one year (minus one day) from the current <code>EditEffectiveDate</code> of this policy transaction. PolicyCenter calculates the expiration date then stores it in the <code>ApprovalInvalidFrom</code> field on the <code>UWIssueApproval</code> object.
Three Years	The approval expires three years (minus one day) from the current <code>EditEffectiveDate</code> of this policy transaction. PolicyCenter calculates the expiration date then stores it in the <code>ApprovalInvalidFrom</code> field on the <code>UWIssueApproval</code> object.
Rescinded	Valid until the approval is rescinded. The approval does not expire on its own.

When determining the validity of the approval, PolicyCenter looks at the effective date of the current policy transaction and of future policy transactions. For example, suppose you have a one year approval at the beginning of a six month policy. This approval extends through any policy changes in the current term, the six month renewal, and any policy changes in the following term. It does not cover a second renewal.

This specifies the value of the **DurationType** property on new approvals. The `UWApprovalDurationType` typelist defines these values.

Default values for approval

The underwriting rule specifies default approval values for the associated underwriting issue. PolicyCenter uses these default values in the following ways:

- As an initial setting in the user interface for approvals that do not already have an approval present
- As automated approvals for auto-approvable issues and issues in automated processes
- For determining availability of the **Approve** button, which indicates that you can approve a specific issue to the default level

PolicyCenter creates approvals that use the default values for auto-approvable issues and automated processes, such as automatic renewal. All blocking points except **Non-blocking** can stop progress of the job pending manual approval. You need to carefully consider the impact of setting the default approval to block or not to block.

In the Underwriting Rule, the default approval fields are:

- Default Duration
- Default Approval Blocking Point
- Default Edit Before Bind
- Default Value Assignment Type
- Default Value Offset Amount
- Default Value Offset Currency

Underwriting Rules screen

Underwriting rules define the circumstances that trigger underwriting issues in a policy transaction. You define underwriting rules in the **Administration**→**Business Settings**→**Business Rules**→**Underwriting Rules** screen.

To view and edit the **Underwriting Rules** screens requires specific permissions.

See also

- *System Administration Guide*

Externally managed rules

In the list of rules,  in the **State** column indicates whether the rule is externally managed or invalid. For externally managed rules, the rule condition is defined outside business rules, often in Gosu code.

See also

- *Configuration Guide*

Clone business rules

About this task

You can use cloning to copy an existing rule and use it as the basis for a new rule.

Procedure

1. Open the rule in PolicyCenter by selecting **Administration**→**Business Settings**→**Business Rules**→**Underwriting Rules**. PolicyCenter displays list of existing rules by name.
2. Select the rule in the list by using the check box in the column to the left.
3. Click **Clone**.

A new rule is created and is available in the **Rule** screen to edit.

Promote business rules

About this task

Once a draft rule is complete and has been validated, it is ready to be moved to a staging area for evaluation. This is typically done in the development environment.

Procedure

1. Open the rule in PolicyCenter by selecting **Administration**→**Business Settings**→**Business Rules**→**Underwriting Rules**. Existing rules are listed by name.

2. Click the rule name to open it.
3. In the rule details screen, select **Promote to Staged**.
The rule status in the **Version** field is now changed to **Staged**.

See also

- “Business rule states” on page 644

View business rule history

About this task

Changes to rules are recorded in the rule **History**.

Procedure

1. Open the rule in PolicyCenter by selecting **Administration**→**Business**→**Settings**→**Business**→**Rules**→**Underwriting Rules**.
PolicyCenter displays list of existing rules by name.
2. Click the rule in the list to open the **Rule** screen.
3. Click **View History**.
The **History** screen displays a list of changes with associated details including the date and time of the change, the name of the responsible user, and the system name. It also provides details on import status, if any.

Enabling or disabling a business rule

PolicyCenter provides the following means to enable or disable a business rule:

- From the rule list screen, by selecting **More**→**Enable Selected** or **More**→**Disable Selected**.
- From the rule details screen, by clicking either **Enable** or **Disable**.
- From the rule details screen, by placing the rule in edit mode and updating the **Enabled** check box.

Whether a user is able to view and use the **Enable / Disable** functionality depends on several factors:

- If the value of configuration parameter **BizRulesDeploymentEnabled** is **true**:
 - The user must have the permission to edit, approve, and deploy business rules.
 - The selected rule version must be the currently deployed rule version.
- If the value of configuration parameter **BizRulesDeploymentEnabled** is **false**:
 - The user must have permission to edit business rules.
 - The selected rule version must be the latest rule version.

Clicking the **Enable / Disable** button on a rule details screen toggles the **Enabled** field on the selected version of the rule. The label of the button changes depending on whether the selected rule version is currently enabled or disabled.

If the selected rule version is the deployed version of the rule and **BizRulesDeploymentEnabled** is **true**, clicking the **Enable / Disable** button does the following:

- Creates a new draft version of the rule.
- Promotes the rule version from draft to deployed automatically.

If the selected rule version is not the deployed version, clicking the **Enable / Disable** button does the following:

- Creates a new draft version of the rule.
- Toggles the **Enabled** field on the rule.

To be absolutely clear, if the rule version is already deployed, then enabling or disabling the deployed rule creates another deployed version of the rule, except with the new status.

See also

- *Configuration Guide*

Deleting a business rule version

PolicyCenter associates a version number and a status (state) with each individual business rule. Whether it is possible to delete any given version of a rule depends on the state of the business rule. The following table lists the rules for deleting a rule version.

Rule version state	Rule deletion action
Deployed	It is not possible delete a rule version after its deployment
Approved or Staged	Clicking Delete deletes all rule versions down to the last deployed rule version.
Draft – Direct parent deployed	Clicking Delete Draft deletes all rule versions down to the last deployed rule version.
Draft – Rule previously staged or approved	Clicking Delete Draft deletes the immediate draft version of the rule. After deleting the draft version of the rule, it then becomes possible to delete the staged and approved versions of the rule.

Underwriting Rule screen

You edit or add a new underwriting rule on the underwriting Rule screen.

The following items appear at the top of the screen:

Item	Description
Edit	Edit this rule.
Delete	Delete this rule.
Delete Draft	Delete this rule which is in draft status.
Promote to Staged	Promote this version to staged.
Validation errors	Displays one of the following: <ul style="list-style-type: none"> • Hide validation errors • Show validation errors • There are no validation errors You can toggle between showing and hiding validation errors.
Version	The version number starting at 0. The version increases when deployed to a production server. Includes the status, Draft or Staged , in parentheses. Version numbers that show a digit only indicate deployed versions of a rule. Version numbers that show + with status indicated non-deployed work-in-progress rules.

Below **Edit** and other buttons, a message appears if a rule is externally managed. Externally managed rules have their rule condition defined outside business rules, often in Gosu code. You can only edit fields related to the underwriting issue type.

See also

- *System Administration Guide*

Rule Details tab on Underwriting Rule screen

The **Rule Details** tab has the following fields. Some of these fields correspond to properties on the **UWIssueType** object.

Field	Description
Name	Specify a name for the issue type. The name identifies the rule in the user interface. Corresponds to the <code>UWIssueType.Name</code> property.
Code	Identify the issue type. The code must be unique among all rules. Corresponds to the <code>UWIssueType.Code</code> property.
Checking Set	Specify when to check for existence or absence of an underwriting rule. The evaluator classes further enforce which jobs check for that rule. In some cases, PolicyCenter uses this value to filter rules for display in the user interface. For checking set values, see “Checking sets and blocking points for underwriting issues” on page 649. Corresponds to the <code>UWIssueType.CheckingSet</code> property.
Blocking Point	Specify the point at which this issue blocks progress of a job. For blocking point values, see “Checking sets and blocking points for underwriting issues” on page 649. Corresponds to the <code>UWIssueType.BlockingPoint</code> property.
Default Duration	Specify the default duration on new approvals. Use this property to indicate when an approval will expire. For duration values, see “Approvals of underwriting issues” on page 653. Corresponds to the <code>UWIssueType.DefaultDurationType</code> property.
Enabled	Specify whether the rule will run in the current environment. Values are: <ul style="list-style-type: none"> • Yes – Rule runs in this environment. • No – Rule will not run regardless of environment.
Last Edited	The user name and date on the last edit.
Global Version ID	A version identifier generated by PolicyCenter that identifies a rule. A rule can be edited over time and produce multiple versions. All versions of that rule have the same identifier. You can use the identifier to determine whether two versions of a rule, maybe even on different servers, are versions of the same rule.
Description	Specify a description of the rule. Corresponds to the <code>UWIssueType.Description</code> property.
Start Date	The start date is compared to the current date to determine whether or not the underwriting rule is in force. This functionality is implemented in the <code>UWRuleApplicabilityCriteriaFilter.gs</code> class.
End Date	The end date is compared to the current date to determine whether or not the underwriting rule is in force. This functionality is implemented in the <code>UWRuleApplicabilityCriteriaFilter.gs</code> class.

See also

- *System Administration Guide*

Applies to section

In the **Applies to** tab you can specify which policy lines, jurisdictions, and policy transactions the rule applies to. You also specify effective start and end dates for the rule. If you specify no dates, the rule is in effect.

You can specify that the rule applies to all policy transactions. Or you can select from a limited list of policy transaction types: **Issuance**, **Policy Change**, **Renewal**, **Rewrite** (includes Rewrite New Account), and **Submission**. If you do not want the rule to be restricted by policy transaction type, then select **All**.

See also

- *Configuration Guide*

Rule context

Specify the **Context** for the **Rule Condition** and **Underwriting Issue Details**. The context determines the autocomplete selections available in formula and text template modes in **Rule Condition** and **Underwriting Issue Details**. The contexts defined in the base configuration include:

- **Generic Policy** – Provides objects for all policy types.
- Policy line – One or more contexts for each policy line.

Rule condition editor

Enter additional conditions which must be met to raise the underwriting issue. For example, raise an underwriting issue if the driver is under 21 years of age. For more information, see “Specifying business rule conditions” on page 665.

Underwriting issue details section

Enter details for underwriting issues that this rule creates when the rule condition is met. Each underwriting issue has an issue key identifier, a value, and short and long descriptions. These fields correspond to properties on the **UWIssue** object.

For more information, see “Specifying underwriting issue details” on page 667.

Advanced Info tab on Underwriting Rule screen

This topic describes advanced settings for underwriting rules. These fields correspond to properties on the **UWIssueType** object. Some of these fields specify default approval values for approving issues.

Field	Description
Default Approval Blocking Point	Specify the default approval blocking point. This value must be at least one level higher than the Blocking Point , or both must be Non-Blocking . This column has the same values as Blocking Point . For values, see “Checking sets and blocking points for underwriting issues” on page 649. Corresponds to the UWIssueType.DefaultApprovalBlockingPoint property.
Value Comparator	Specify how to compare the issue value. You can specify: <ul style="list-style-type: none">• If the value of an issue is within the authority granted to a user• If value of an approval is within the authority granted to a user• If the value of an issue is within the associated value of the approval For values, see “Business rule: value comparator values” on page 647. Corresponds to the UWIssueType.Comparator property.
Default Edit Before Bind	Specify the default value for new approvals. 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 job. Corresponds to the UWIssueType.DefaultEditBeforeBind property.
Auto-approvable	Specify whether this issue is treated as a auto-approvable issue. Auto-approvable issues can be automatically approved if the user has an authority grant that permits an approval at the default level. See “Approvals of underwriting issues” on page 653. Corresponds to UWIssueType.AutoApprovable property.
Default Value Assignment Type	Specify how to compute a default approval value from the value of the issue. PolicyCenter uses this value if Value Comparator is At least or At most . For values, see “Business rule: default value assignment type” on page 648. Corresponds to the UWIssueType.DefaultValueAssignmentType property.
Default Value Offset Amount	Positive values produce a default approval value that is somewhat riskier than the issue value. Use if Default Value Assignment Type is Offset Amount or Offset Percentage . If Default Value Assignment Type is Offset Amount or Offset Percentage , this field is the associated value or percentage to use in the offset calculation. Corresponds to the UWIssueType.DefaultValueOffsetAmount property.
Value Formatter Type	Define which output formatter formats the value. For values, see “Business rule: value formatter type” on page 648.

Field	Description
	Corresponds to the <code>UWIssueType.ValueFormatterType</code> property.
Default Value Offset Currency	Use if Default Value Assignment Type is Offset Amount . This field sets the default currency to use in the offset calculation.
cy	Corresponds to the <code>UWIssueType.DefaultValueOffsetCurrency</code> property.

Entering expressions in business rules

In the **Rule Condition**, expressions appear on the left and right of a condition row.

Expressions can be in one of the following modes:

Mode	Description
Formula	Write an expression using literals, comparison operators, and object properties.
Count	In a list, count the number of items matching a given condition.
Sum	In a list, sum up the expression for every item matching a given condition.
Lookup	Creates an expression that returns a value from a data lookup table.

Select the mode by clicking  to the immediate right of an expression.

Formula expressions in business rules

In formula expressions, you enter literals, context objects, arithmetic operators, and utility functions directly in the text field. The arithmetic operators are +, -, *, /, and %. For information about these operators, see the *Gosu Reference Guide*. Utility functions are defined globally and are always available.

When you start typing, autocomplete shows you available objects, properties, and methods based on the selected **Rule Context** and filtered by the text you enter.

To select from available objects, press Spacebar. Type one or two letters to display first-level objects containing letters in that order. Type a period after the object to select its subobjects, properties, and methods. Type three or more letters to display objects containing letters in that order. The letters can match objects at multiple levels of nesting. The number of levels is defined in the `BizRulesLeafSearchNumOfHops` parameter in `config.xml`. The default value is 3.

For example, if you type cont, you can choose objects at three levels such as `policyEvalContext`, `policyPeriod.BillingContact`, and `policyPeriod.BasedOn.ContainerId`.

To access utility functions, type `Util` followed by a period.

The following formula expressions use literals and comparison operators:

- 50
- 50 + 50
- "some" + "thing"

Enter formula expressions in business rules

About this task

The following formula expression uses objects, literals, and comparison operators:

- `policyDrivers.Count / vehicles.Count * 100`

To enter this formula expression:

Procedure

1. In Administration→Business Rules→UW Rules, edit the **PA: Number of vehicles** underwriting rule.
2. In Rule Condition, click Add.
3. Set the mode to Formula by clicking  to the immediate right of an expression.
4. In the **Left Expression**, press Spacebar to view available objects, and then select **policyDrivers**.
5. Type a period to view objects and properties on the object.
6. Select **Count**.
7. Type / for the division operator.
8. Type v to view all objects and properties containing v, and then select **vehicles**.
9. On **vehicles**, select the **Count** property.
10. Enter the * operator followed by the literal 100.

Count and sum expressions in business rules

You enter both count and sum expression types in the same way. These expressions can include a Boolean condition to filter the list.

The following are examples of count and sum expressions:

- Sum the new vehicle cost of all convertibles, coupe, and four-door sedans:

```
the sum of  
    vehicle.CostNew for each personalVehicle in the vehicles list  
    where vehicle.BodyType Is In "Convertible", "Coupe", "Four-door sedan"  
• Count the number of drivers over the age of 18:  
The count of  
    each policyDriver in the policyDrivers list  
    where policyDriver.Age > 18
```

Enter count and sum expressions in business rules

About this task

To enter a count or sum expression:

Procedure

1. In Rule Condition, click Add.
2. To the right of the expression, click  and select **Count** or **Sum**.
For this example, select **Count**.
3. Click **Set fields and condition** to display the **Count** or **Sum** popup screen.
4. In **In This List**, type space to select from available objects. When you start typing, autocomplete shows you available objects, properties, and methods based on the rule context and filtered by the text you enter. The expression must evaluate to a list.
For this example, select the list **policyDrivers**.
5. Select the type of count, and specify conditions of items to count.

Below the condition table, the formatted view displays the complete rule condition with the count or sum that you are editing highlighted in yellow.

For this example, set the condition to:

```
policyDriver.Age > 18
```

Lookup functions in data lookup tables

You can query data lookup tables in business rules using *lookup functions*.

You can create data lookup tables from any system table. The data lookup table includes all or part of the system table data and a lookup function.

Lookup functions takes one or more inputs and query a data lookup table. There are two types of lookup functions. One type returns a Boolean value indicating whether or not the input values exist in the table. The other type returns a single value from the table. In business rules, you can use lookup functions in the following places:

- Rule variable expressions
- Left or right expressions in the condition builder

You create a lookup function on the **Create New Lookup** screen. You can access this screen in several ways, including the following:

Screen	Action to take
Manage Data Lookup Tables screen	<ol style="list-style-type: none"> 1. Navigate to the following location in PolicyCenter: Administration→Business Settings→Business Rules→Manage Data Lookup Tables 2. Click Add Lookup.
Select the Data Lookup screen	<p>In a business rule Condition pane, select the Lookup option for an expression.</p> <ol style="list-style-type: none"> 1. Click Set lookup to display the Select the Data Lookup screen. 2. Click Create New Lookup.

Example: Return start and end dates

In a business rule, you need to know whether the policy effective date is between the start and end dates.

You have a data lookup table with these columns:

State	Start date	End date
AL	07/01/2018	12/31/2018
NM	06/01/2018	11/31/2018
...		

One lookup function queries a data lookup table for the start date for a particular U.S. state. Another lookup function queries for the end date.

Without lookup functions, you need 50 expressions to get the start date for all 50 U.S. states, and 50 more to get the end date. With lookup functions, you need only two expressions.

Example: Return an amount

In a business rule, you need to know whether a particular amount on the policy surpasses a threshold set for a particular state and line of business (LOB).

You have a data lookup table with these columns:

State	LOB	\$ Threshold
Colorado	PersonalAuto	1000
Colorado	CommercialAuto	1500
Wyoming	PersonalAuto	900
...		

You can use a lookup function to get the threshold for a particular state and line of business (LOB). Your rule condition can check whether the amount on the policy surpasses the threshold.

Example: Check for existence

In a business rule, you need to check whether you can issue a homeowners policy if the homeowner has a particular dog breed.

You have a data lookup table that lists the dog breeds that cannot be on a policy.

Dog breed
Doberman
Rottweiler
Bull Terrier

You can use a lookup function to return a Boolean indicating whether a particular dog breed is listed in the table.

See also

- “Manage Data Lookup Tables screen” on page 668

Use a lookup function

About this task

Follow these steps to use a lookup function in an underwriting rule. The steps create a new rule, but you can modify an existing rule as well.

Procedure

1. In PolicyCenter, navigate to the following screen:
Administration→Business Settings→Business Rules→Undewriting Rules
2. On the **Underwriting Rules** screen, click **Add** to add a new rule.
3. For the purposes of this example, do the following:
 - a. Under **Applies To**, choose **Selected in Policy Lines**.
 - b. Select **Homeowners** and click the right arrow.
4. Do the following as appropriate:
 - To use a lookup function in **Variables**, select **Lookup** from the **Expression** drop-down list.
 - To use a lookup function in a **Condition**, select **Lookup** from the **Right Expression** or **Left Expression** drop-down list.
5. Click **Set lookup** to display the **Select the Data Lookup** popup.
 - a. Select a table under **Table Name** to view the **Lookup Table Details**. For the purposes of this example, select **Ineligible Dog Breeds**.
 - b. In **Lookup Table Details**, enter the following code in the **Expression** field for **DogBreed**:
`hopDwellAnimal.AnimalBreed.DisplayName`
Guided editing helps you enter this expression.

Working with business rule variables

It is possible to define a variable in PolicyCenter business rules that stores a specific expression or value for later reuse in the rule. In this manner, you define an expression once and then reuse it at will in the conditions or actions of the rule. Thus, rule variables are similar to rule symbols. Just as you can use symbols defined in the context definition in the rule conditions and actions, you can use variables in the rule conditions and actions as well.

Adding a rule variable

Guidewire recommends that you first set the rule context before defining the rule variable (as you must with condition expressions as well). This ensures that the rule editor recognizes any rule symbol that you use in the variable expression.

To add a rule variable to a rule, place the rule in edit mode and click **Add** in the **Variables** section of the editor. It is possible to add multiple variable rows in the **Rule Variables** section by clicking **Add** multiple times. However, if you do so, PolicyCenter removes any variable row that is completely empty during a save operation of the rule edits.

After you add a rule variable row, enter the following information.

Field	Required	Description
Name	Yes	Use alpha-numeric characters only. The variable name must start with a letter and meet Java naming conventions. The variable name cannot be a reserved Gosu or Java keyword. Each variable name must be unique within the entire variable name space.
Description	No	A simple sentence that describes the meaning and purpose of the variable.
Expression	Yes	<p>Expressions that you enter in this field must one of the following forms:</p> <ul style="list-style-type: none"> • Formula • Count • Sum • Lookup <p>Use the drop-down picker to set the form of the expression. If shown, click the link in the expression to open up a secondary screen in which you must enter additional information.</p>
Type		PolicyCenter populates this read-only field automatically after you enter a value in the Expression field. This field shows the return type of the defined expression.

After you define the rule variable values, and after PolicyCenter determines the variable name and expression to be valid, PolicyCenter adds the rule variable to the rule context definition.

Each rule variable that you define must have a unique name across the entire variable name space. However, a variable that you define in a rule is only available to that rule, not across other rules. A variable that you define in a particular rule version is not available to prior versions of that rule. Thus, rule versioning, rule lifecycle, rule deployment, and rule import and export all apply to rule variables as well.

For more information about expressions, see the following topics:

- “Formula expressions in business rules” on page 660
- “Count and sum expressions in business rules” on page 661
- “Lookup functions in data lookup tables” on page 662

Working with rule variables

Rule variables work in exactly the same manner as the other symbols in the context definition. This means that you can perform such operations as `var1 + var2` and other similar operations.

In working with rule variables:

- It is possible to use variable expressions of any form in rule condition and action statements, on either side of the line.
- It is possible to use rule variables in underwriting rule action parameters that permit Gosu expressions. Use the `$ {}` construct to enter a variable into an action parameter.
- It is not possible to nest one variable inside another variable. (A variable expression cannot itself contain a previously defined variable.) However, it is possible to use variables in any of the nested expressions (sum, count, lookup) if accessed through the condition builder.
- It is not possible to use rule variables whose form is Sum or Count within underwriting rule action statements.

PolicyCenter invalidates the entire rule if a rule variable is invalid.

Removing a rule variable

If you use a variable in a condition line and then delete the variable definition, PolicyCenter generates a validation error. If you rename a variable, PolicyCenter again generates a validation error if the variable exists in a rule condition line.

Lookup expressions and rule import

During rule import, if a rule has a lookup expression defined to a lookup table that does not exist in the target environment, PolicyCenter marks the rule as invalid. The import of the rule still succeeds. However, you must make the necessary changes indicated in the **Rule Detail** screen to make the rule valid before it can execute in the target environment.

Specifying business rule conditions

On the **Rule** screen in **Condition**, you specify additional conditions that must be met for this rule to create an underwriting issue. In new rules, the default condition is **AND**, which means the rule fires if all the applicability criteria are met. You can replace this condition with your own.

The condition type can be one of the following:

- **None** – Always fires when the rule is executed.
- **All of the following criteria must be true (AND)** – All rows evaluate to true.
- **At least one of the following must be true (OR)** – At least one row evaluates to true.
- **The following combination of criteria must evaluate to true (AND/OR)** – Multiple rows can be combined by using AND, OR, and parentheses for grouping. The rows evaluate to true.

Each row requires a **Left Expression** followed by an **Operation**. The **Right Expression** is required except when the **Operation** is **Is True**, **Is False**, **Has Value**, or **Has No Value**.

So that you can more easily comprehend the condition, PolicyCenter displays a formatted view of the condition below the condition table.

Note: Avoid commas when using numerical values in business rules expressions. Use commas only in lists.

The following are examples of conditions:

- Example of AND:

```
vehicle.VehicleType = "Other"  
AND  
vehicle.Vin.startsWith("FRE") Is True
```

- Example of AND/OR:

```
vehicle.Vin.startsWith("FRE") Is True  
OR  
( vehicle.PrimaryDriver.PolicyDriver.DateOfBirth Is Not Equal To null  
AND  
vehicle.PrimaryDriver.PolicyDriver.Age < 25 )
```

Operations in business rule conditions

Operations appear between the left and right expressions. In rule conditions, operations specify how the left expression is compared against the right expression. Each rule condition row requires an operation. There are three types of operations:

- Comparison
- Unary
- Functional

Comparison operations

Comparison operations require both left and right expressions. Both expressions must evaluate to the same type. The comparison operations are:

Operation	Description
=	Left expression is equal to the right expression.
Is Not Equal To	Left expression is not equal to the right expression.
<	Left expression is less than the right expression.
<=	Left expression is less than or equal to the right expression.
>	Left expression is greater than the right expression.
>=	Left expression is greater than or equal to the right expression.

The following example conditions use comparison operations:

- `vehicles.length >= 5`
- `vehicle.GarageLocation.PostalCode Is Not Equal To null`

Monetary expressions

When using monetary amount properties in operations in the rule condition builder, drill down to the appropriate `Amount` property to avoid validation errors.

For example:

- `hopDwelling.Branch.PaymentAmount.Amount < 1000.00`

Unary operations

Unary operations require only a left expression. For **Is True** and **Is False**, the expression must evaluate to a Boolean.

Operation	Description
Is True	Left expression is true.
Is False	Left expression is false.
Has a Value	Left expression has a value. The expression does not evaluate to null.
Has No Value	Left expression evaluates to null.

The following example conditions use Boolean operations:

- `policyDrivers.isEmpty() Is True`
- `personalAutoLine.PAPIP_ARExists Is False`

Functional operations

Functional operations require both left and right expressions.

Operation	Description
Is In	Item in the left expression is contained in the list in the right expression.
Is Not In	Item in the left expression is not contained in the list in the right expression.
Contains	List in the left expression contains at least one item matching the condition in the right expression.
Does Not Contain	List in the left expression does not contain any item matching the condition in the right expression.

For **Is In** and **Is Not In**, the right expression must evaluate to a list, and the left expression must evaluate to a type matching an item in the list.

For **Contains** and **Does Not Contain**, the left expression must evaluate to a list, and the right expression specifies a condition that items in the left expression are compared against.

The following conditions use functional operations:

- `policyPeriod.BaseState Is In "California", "Oregon", "Washington"`
- `policyPeriod.AllCosts Contains a cost where cost.ActualAmount >= cost.ActualTermAmount`

Specifying underwriting issue details

At the bottom of the **Rule** screen, you can specify **Underwriting Issue Details** for the underwriting issues that this rule creates. Specify an issue key identifier, a value, and short and long descriptions. These fields correspond to properties on the `UWIssue` object.

Field	Description
Issue Key	The underwriting rule and issue key uniquely identify each underwriting issue. For any given combination of underwriting rule and issue key, there can be only one issue on a policy period at a given time. When creating an issue, any pre-existing issue with the given type and key will be returned, and any inactive issue that matches will be activated and then returned. For example, if only one underwriting issue of this type can appear on a transaction, just specify the name or code of the rule. If multiple issues of the same type can appear on a transaction, choose the value that distinguishes one item from another, such as vehicle identification number or class code. Corresponds to the <code>UWIssue.IssueKey</code> property. For configuration information, see the <i>Configuration Guide</i> .
Value	Specify the value for this underwriting issue. Corresponds to the <code>UWIssue.Value</code> property.
Short Description	Specify a short description for this underwriting issue. Corresponds to the <code>UWIssue.ShortDescription</code> property.
Long Description	Specify a long description for this underwriting issue. Corresponds to the <code>UWIssue.LongDescription</code> property.

For each field, click  to specify whether the field is a **Text Template** or **Formula**. If a field is a formula,  appears after the field name.

Text Template

If the field is a text template, enter text. For example:

- This policy requires additional approval.

The text can also contain a formula. To include a formula, embed the expression between `{}$`. After typing `{}$`, autocomplete shows you available objects, properties, and methods based on the context selected for the rule being edited and filtered by the text you enter. Type space to select from available objects. Type one or more letters to display objects containing letters in that order. Type a period after the object to select its subobjects, properties, and methods.

An example of a text template:

- This policy in `{}${policyPeriod.BaseState}` requires additional approval.

See also

- *Gosu Reference Guide*

Formula

In formula expressions, you can enter literals, context objects, arithmetic operators, and library functions directly in the text field.

See also

- “Formula expressions in business rules” on page 660

Managing business rule export and import

You transfer rules between different server environments by exporting the rules from one server and importing the rules into the other server. Typically, dedicated Rule Administrators manage the export and import of business rules between different PolicyCenter servers.

During the rule import process, it is possible for rule conflicts to occur. This can happen, for example, if you import a different version of a rule that already exists on the importing server. PolicyCenter detects and highlights these types of issues. The Rule Administrator manages these rule conflicts manually, in the **Business Rules** import screens.

IMPORTANT The user who is responsible for resolving rule conflicts, either in a production environment or non-production environment, must have the necessary rule edit permission.

To access the main **Business Rules** import and export screen, navigate to the following location in Guidewire PolicyCenter:

Administration→**Business Settings**→**Business Rules**→**Underwriting Rules**→**Import/Export Status**

Data lookup tables must be exported and imported separately from business rules. You must import the data lookup tables before importing the business rules that use them.

See also

- *System Administration Guide*

Exporting and importing data lookup tables

You can transfer data lookup tables between different server environments by exporting them from one server and importing them into another server. To export, navigate to **Administration**→**Utilities** and select **Export Data**. Select **Lookup Table Definitions**. The data is exported to an XML file.

You can import the XML file containing data lookup tables on the **Import Data** screen. When importing, the data can conflict with existing data. You can choose to overwrite or discard all conflict changes. You can also choose to how to handle each data lookup table individually.

You must import data lookup tables before you import the underwriting rules that use them.

Exporting and importing underwriting issue types

You can transfer underwriting issue types between different server environments by exporting them from one server and importing them into another server.

To export, navigate to **Administration**→**Utilities** and select **Export Data**. Select **Underwriter Issue Types**. The data is exported to an XML file that you can import on the **Import Data** screen.

Manage Data Lookup Tables screen

The **Manage Data Lookup Tables** screen displays the data lookup table definitions that have been added to the system. From this screen, click **View** or **Edit** to view or edit a data lookup table definition, including its lookup function. Click **Add Lookup** to add a new data lookup table definition.

You cannot delete or edit a data lookup table that is referenced by an enabled rule.

Create New Lookup screen

Use the **Create New Lookup** screen to define a new data lookup table definition, which includes the lookup function. Unless marked as optional, all fields require a value.

Field	Description
Source Table	Data lookup table to query for data. Use the drop-down picker to select the table. After you select a table, PolicyCenter shows the names of the table columns and other information about the table at the bottom of the screen.
Name	Name for the lookup function.
Description	Optional description of the lookup function.
Table Code	Hash code for data lookup table to query. PolicyCenter populates this field automatically.
Existence	Boolean check box field with the following meaning: <ul style="list-style-type: none">• Yes - Checks for the existence of a value in the designated input columns.• No - Returns the actual value from the column designated as the output column.
Table columns	Use to select the table column or columns to include in the lookup function.

Setting the Existence field

After you select the table to query, you must then set the query **Existence** field. This field presents a Boolean choice of **Yes** or **No**. The following table summarizes the functionality of the **Existence** field.

Existence	Table columns	Returns
Yes	For existence-type lookup functions. Do not designate an output value. All columns are inputs.	Boolean true or false depending on whether values exist in all table columns marked as inputs.
No	For non-existence lookup functions. Designate one, and only one, column as an output value.	Actual value stored in the table column marked as the output to the query.

Setting the Input and Output columns

After you set the **Existence** field, you need to determine which table columns you want to be part of the query. PolicyCenter selects all table columns automatically for inclusion in the lookup function by default. To remove a column, select its check box and click **Remove Column**.

For the remaining table columns, you need to decide whether they are input or output columns, depending on how you set the **Existence** field.

- If you set **Existence** to **Yes**, you must set all table columns to **Input**. If you attempt to do otherwise, PolicyCenter displays a validation error.
- If you set **Existence** to **No**, you must set all but one of the table columns to **Input**. If you do not set one, and only one, table column to **Output**, PolicyCenter displays a validation error.

You can set a table column to either **Input** or **Output**, but not both values simultaneously.

Underwriting issues

PolicyCenter raises *underwriting issues* when a policy transaction contains items that may require underwriter review. Underwriting issues are part of underwriting authority.

Underwriting issues overview

PolicyCenter raises underwriting issues at defined points in a policy transaction. For example, if a personal vehicle policy transaction contains more than five vehicles, PolicyCenter raises an underwriting issue before quoting. If an underwriter does not approve this issue, the policy transaction cannot be bound.

Underwriting issues are raised based on a specific condition on the policy transaction, without regard to who the user is who is entering it. In PolicyCenter, underwriting authority allows authorized underwriters to automatically approve underwriting issues so that their work is not impeded, but the issue still appears on the transaction.

Each underwriting issue is raised because of an underwriting rule. The sample data contains sample underwriting rules. You can create your own underwriting rules. For example, the sample data contains a PA: Number of vehicles underwriting rule for the Personal Auto policy line. Underwriting rules define when to raise an underwriting issue. This underwriting rule raises an issue if the number of vehicles exceeds a certain number.

Working with underwriting issues

This topic describes how to work with underwriting issues in the PolicyCenter user interface.

Underwriting 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. You can add issues if the underwriting rule has Checking Set of Manual.
Request Approval	Request approval from another user, such as an underwriter. Takes you to the UW Activity screen where you create an activity for the other user 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.

Button	Description
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** drop-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. However, if you have the <code>uwreopen</code> permission, this button appears for all approved and rejected issues regardless of which user approved or rejected them.

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.

Risk Approval Details popup for risk analysis

When you click **Approve** on the **Risk Analysis** screen, PolicyCenter displays the **Risk Approval Details** popup. The default values for the approval are defined in the underwriting rule. 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.

Field	Description
Allow Edit	Initial value of this radio button comes from Default Edit Before Bind on the Advanced Info tab of the underwriting rule. 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 Default Approval Blocking Point on the Advanced Info tab of the underwriting rule. 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.
Valid until	The initial value for this drop-down list is from Duration of the underwriting rule. 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 – Valid until the approval is rescinded. The approval does not expire on its own.
Offset Percentage	For value issues, if the Default Value Assignment Type of the underwriting rule is set to Offset Percentage , then use the Default Value Offset Percentage . The default approval amount in Reference Value 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 Default Value Assignment Type of the underwriting rule is set to Offset Amount , then use the Default Value Offset Amount . The default approval amount in the Reference Value is for the number of the item plus (or minus) the offset amount. For example, the offset amount for the Number of Vehicles rule is 1 and the Value 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

- “Authority profiles” on page 721 for more information about authority profiles in the PolicyCenter application and how to work with them.

History of underwriting decisions

PolicyCenter displays the history on the **Risk Approval Details** screen displays the history for each underwriting issue. The history includes details such as:

- Approver
- Date of the approval
- Effective date
- 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.

Add underwriting referral reasons

About this task

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 for that policy.

Procedure

1. Go to an account and select a policy.
2. Click **Risk Analysis** in the left sidebar.
3. Select the **Add UW Referral Reason** button.
4. Select an underwriting rule, and enter a short and long description.

The **UW Referral Reasons** tab displays all underwriting referral reasons on the policy.

You can close a underwriting referral reason, open a closed one, or view details.

Part 9

Documents

Document management

PolicyCenter enables you to create and manage documents that are associated with accounts and policies. These documents have content that exists in or is created in PolicyCenter. For example:

- You write and send the insured a letter to acknowledge a new submission.
- The insured emails you documents related to the safety of the insured location or garage.

You can use documents for generating and tracking information that is not part of the policy contract. For information that is part of the policy contract, use policy forms instead.

Guidewire recommends integrating with an external document management system rather than using the default demonstration document management system on the PolicyCenter server. The default system is useful only for demonstration purposes and does not support features of a real document management system, such as document versioning.

Use document management in PolicyCenter to:

- Create new documents on the server from templates, and then download and edit them.
- 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.
- Remove documents.
- Associate a document with an account, policy, or contingency.
- Create and send a document from rules or workflows.
- Extend these default capabilities by integrating with an external document management system (DMS).

By default, PolicyCenter stores document contents as files on your PolicyCenter server. For more robust document management, integrate documents with an external document management system.

See also

- “Policy forms” on page 435
- *Globalization Guide*
- *Integration Guide*

Document storage overview

This topic describes how PolicyCenter stores documents as configured in the base product. You can configure how PolicyCenter uses metadata properties and stores content.

In the base configuration, documents are stored as a combination of:

Metadata

Properties that specify information about a document. In the base configuration, PolicyCenter stores these properties in the database. For example, there are properties for the document's name, the business object associated with the document, the document's file type, an optional type classification, and so on. When you create a new document, you must specify some of its properties before you can save it.

For example, you see document properties when you click the Info action  for a document in the **Documents** screen.

Document content

A file that is stored in the PolicyCenter file system. In general, you edit document content as a file on your local system by using your editing software. Alternatively, you can create the file from a template and, in some cases, edit that file on your local system. Before uploading the content, you select or specify the metadata representing the document in PolicyCenter. You then upload the file to the server, which associates the file with its metadata and saves the file.

For example, you can view a document's content by clicking the document name in the **Documents** screen.

See also

- “Document metadata properties” on page 678
- “Viewing documents” on page 679
- For information on configuring document storage, see the *Integration Guide*.

Document metadata properties

When you create a new document or edit an existing document, you see a set of metadata properties that the base configuration of PolicyCenter stores in the database. Document search uses a subset of these properties.

You can set the following metadata properties for a document:

Name

The name of the document. PolicyCenter uses this name for the document content file. For example, if you download the content for a document, this setting determines the name of the file name sent to the browser.

Description

Especially useful for locating hard-copy documents.

File Type

The type of content file, also known as a MIME type.

You can change the file type, but do so with caution. PolicyCenter uses your setting to set the MIME type for the file. The operating system formats the document content file to match this MIME type when you upload the content.

Author

By default, the name of the user who associated the document with the account, policy, or contingency. This field can be changed to some other value, such as the sender of a document.

Recipient

The person or business to which the document was sent, if applicable.

Language

The language the document is written in.

Related To

The account, policy, or policy transactions that the document is related to.

Status

A value from the `DocumentStatusType` typelist, such as Final or Draft. You are required to set this value when you create a document. In the base configuration, only Final and Draft are used. The Approving and Approved statuses are not used in the base configuration, but you can implement code that uses them.

Security Type

The default values are Internal Only, Sensitive, and Unrestricted. For example, a document might require extra restrictions on users who can view and edit the document.

Document Type

A value from the `DocumentType` typelist that classifies the document, such as Policy summary or Email Sent.

Section

A way to classify documents, such as legal, medical, or correspondence.

Hidden

Indicates whether the document is hidden or visible.

See also

- “Configuration parameters for document management” on page 685
- “Create a new document” on page 681

Working with documents

Open an account or policy and select the **Documents** link in the left Sidebar to work with it. To create a new document, select **New Document** from either the **Actions** menu or the **Documents** screen while in a policy or account screen. Or, you can open a contingency to create new documents or view existing ones for the contingency.

Viewing documents

You can view all documents associated with accounts, policies, or contingencies, or you can filter the list and see a subset of these documents.

You can view all documents for which you have permission.

See also

- For information on document permissions, see “Document security” on page 685.
- For information on document configuration parameters, see “Configuration parameters for document management” on page 685.

Viewing all documents

The **Documents** screen shows all documents associated with an account, policy, or contingency.

To open this screen, open an account or policy and click the **Documents** link, located on the left Sidebar of all policy or account screens.

The **Documents** screen initially displays the unfiltered list of all documents. Use the search pane at the top of the screen to filter the list of documents.

In the list of documents, you can:

- Click a document **Name** to download the document and view its contents.
 - If the browser can open the document for viewing, a window opens showing the contents.
 - If the browser cannot open the document for viewing, you see a message saying that the file was downloaded for viewing. You can then open the downloaded file with the appropriate viewer.

If nothing happens when you click the document name, enable pop-ups for PolicyCenter in your browser.

- Click View Document Properties  to see the document's metadata properties on the **Document Properties** screen. On that screen you can edit the properties, download the document content, or upload new content.
- Click Download  to download, view, and possibly edit the document's content.
- Click Upload  to upload new or edited content.
- Click Delete  to delete the document.

View documents for a contingency

Procedure

1. With a policy open, click Risk Analysis in the Sidebar menu on the left.
2. On the **Risk Analysis** screen, click the **Contingencies** card.
3. Click the title of a contingency to open the **Contingency** screen.
The **Documents** section shows all documents linked to the contingency.
4. You can view a document's contents, see or edit its metadata properties, download and upload document contents, and delete the document.
5. You can use the buttons above the list, **New Document** and **Delete Selected**, to create a new document from a template, upload content, and delete the document.

See also

- For information on using **Name** and **Actions** for viewing content or metadata properties, downloading and uploading content, or deleting a document, see “Viewing all documents” on page 679.
- “Contingencies” on page 349.

Searching for documents

Use the **Search** pane of the **Documents** screen to search for documents. You can use the following search parameter values for a document after you create the document or link to it:

- **Document Name** – The name of the document. Typically, also the name of the file in which the document content is stored.
- **Related To** – When viewed from an account, the drop-down list enables you to filter by the current account, or policies and policy transactions on the account. When viewed from a policy, you can filter by the current policy and policy transactions. Select **<none>** to find all documents related to the current account or policy.
- **Status** – A value from the **DocumentStatusType** typelist, such as Final or Draft. You are required to set this value when you create a document. In the base configuration, only Final and Draft are used. The Approving and Approved statuses are not used in the base configuration, but you can implement code that uses them.
- **Date Range** – The start and end of a date range.
- **Author** – By default, the name of the user who associated the document with the account, policy, or contingency. This field can be changed to some other value, such as the sender of a document.
- **Include Hidden Documents** – Whether to search also for documents that have been hidden.

See also

- “Hiding a document” on page 684
- *Configuration Guide*

Create a new document

Before you begin

You can create a new account or policy document when you have an account or policy open.

You can also create a document:

- For a contingency when you have a policy open. See “View documents for a contingency” on page 680.
- From an email. For example, with an active or pending account open, you select **Actions**→**New Email**. Then in the **Email** worksheet, you set **Save as Document**.
- From an unbound policy’s **Prior Losses** card, available on the **Risk Analysis** screen for the policy.
- From the submission manager when you create a confirmation letter for the submission.

Procedure

1. Either use the **Actions** menu and go to the selections under **New Document** or open the **Documents** screen and click **New Document**.
2. Click one of the following choices for adding documents to the current account or policy:
 - **Upload document**
 - **Create from a template**

See also

- For information on prior losses, see “Risk Analysis screen for personal auto” on page 263.
- “Submission manager” on page 93

Upload documents

About this task

When you upload a document, you replace the content for a document with a file from your file system. If you are creating a new document, you must specify metadata properties for the document, and the upload becomes the content. You can upload multiple documents at one time.

Procedure

1. There are multiple ways to get to the **Upload Documents** worksheet that enables you to upload one or more documents:
 - Click **Actions**, and under **New Document** click **Upload documents**.
 - In the **Documents** window, click **New Document** and then click **Upload documents**.The **Upload Documents** worksheet opens.
2. To add files that you want to upload, do either of the following, or both:
 - Drag one or more files from your file system window, such as Windows Explorer, to the worksheet.
 - Click **Add Files**, browse to the locations of your documents, and click **Add**.
 - You can click **Add Files** multiple times for files in different folders. You can also select more than one document in a folder.
3. Set the properties for the files you want to upload.
 - You must have values for the **Name**, **File Type**, **Related To**, **Status**, **Document Type**, and **Hidden** fields.
 - You can set the properties one file at a time in the fields to the right of each file you added to the list.
 - You can edit the properties for multiple files by selecting their check boxes and then clicking **Edit Details**.
 - Do not set the **Name** field for multiple files. Files must have different names. Additionally, PolicyCenter sets the file type for you based on the MIME type it detects. If you set the **File Type** field, the file contents will be configured to match that MIME type when you upload it.

4. Click **Upload** to send the file or files to the server and create the link or links.

Replace content for an existing document

Procedure

1. You can start the upload to replace a document's content in two ways:
 - On the **Documents** screen, for the document whose contents you want to upload, click **Upload**  under **Actions**.
 - On the **Documents** screen, for the document whose contents you want to upload, click **View Document Properties**  under **Actions**. Then, on the **Document Properties** screen, click **Upload** .
2. In the **Update Document Content** screen, add the file that has the new content by:
 - Browsing for the content file.
 - Dragging the file from your file system window, such as Windows Explorer.
3. Click **Update**.

Create a new document from a template

Procedure

1. Open an account or policy.
2. Select either of the following:
 - **Actions**→**New Document**→**Create from a template**.
 - Open the **Documents** screen, and then click **New Document**→**Create from a template**.
3. In the **New Document** worksheet, click the **Select Template** search icon so you can select a template. To create a document, you must specify an existing template.

When creating a document from a template for a policy, 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.
4. After you click the **Select Template** search icon, a search screen for document templates opens.

The search settings are based on the account or policy you have open.

 - a. If no results are showing, choose a document type from the **Type** picker.

For example, select **Confirmation letter**.
 - b. Set any other search fields that will help you find the template.
 - c. Click **Search**.

The **Search Results** displays a list of matching document templates.
 - d. Click **Select** for the template you want to use.

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.
5. After you select a template, PolicyCenter displays numbered steps along the left side of the screen.
6. Follow the steps on the screen.

The document requires values for **Name**, **Related To**, **Status**, **Document Type**, and **Hidden**. Those values are filled in for you, but you might want to change them. In particular, **Name** sets the file name of the content file.

If you integrate with a document management system, the file attributes used by that system need not be the same as the comparable object values that appear in the document.
7. After filling in the fields, click **Create Document**.

8. If you see **View/Edit**, click this button.
 - If you can edit the document content, your browser will indicate that it downloaded the file.
 - You can use the browser feature that enables you to open the downloaded file in its native editor.
 - If you edit the document content file, be sure to save it.
 - Make note of the saved file name and location so you can browse for the file when you upload changes to the document. The file you upload becomes the new content for the document.
9. Click **Update** to save your work.

Next steps

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, if you have integrated with a document management system, you can use any features provided by that system.

See also

- “Document metadata properties” on page 678

Edit content for a document

About this task

You can edit contents of documents either in the **Documents** screen or in the **Contingency** screen if documents have been added to a contingency.

You can edit the content of a document if you have sufficient permissions.

Note: If you are using Microsoft Internet Explorer and you download a Microsoft Office document, the browser can open it for you in the Office application, such as Microsoft Word. However, it is possible that the browser will not use the correct file name for the document. Before saving a Microsoft Office document that you have downloaded for editing, verify that the file name is correct, and enter it again if necessary.

Procedure

1. Click Download  in the **Actions** column for the document. Alternatively, you can click the same button on the **Document Properties** screen for the document.

Your browser indicates that it downloaded the file.
2. Edit the document content file in the appropriate editor.

Most web browsers can be configured to open some types of downloaded files in their native editors.
3. Save your work after you have made all your edits.

Make note of the saved file name and location so you can browse for the file when you upload changes to the document. The file you upload becomes the new content for the document.
4. In the **Documents** screen, click **Upload**  under **Actions**.
5. On the **Update Document Content** screen, click **Browse**, locate the file you saved, and then click **Update**. Alternatively, you can drag a file from your file system viewer to this screen.

See also

- “Upload documents” on page 681
- “Document storage overview” on page 677

Edit metadata properties of a document

Before you begin

You can edit the metadata properties of a document if you have sufficient permissions.

About this task

You can edit metadata properties of documents either in the **Documents** screen or in the **Contingency** screen if documents have been added to a contingency.

Procedure

1. Click View Document Properties  in the **Actions** column for the document.
2. In the **Document Properties** screen, click **Edit**.
3. Make your changes.
If you change the **Name** field, PolicyCenter subsequently uses that name for the file it downloads for document content.
4. Click **Update** when you have made all your changes.

See also

- “Upload documents” on page 681
- “Document storage overview” on page 677

Hiding a document

Hiding a document is a way to remove an obsolete document from your list of documents without deleting it. When you hide a document, you no longer see it listed in the **Documents** screen unless you indicate that you want to see hidden documents.

You can hide a document in a number of ways:

- With an account or policy open, open the **Documents** screen from the left Sidebar, select a listed document, and click **Hide Documents**.
- Open the **Documents** screen. Then click View Document Properties  for a document to open its **Document Properties** screen, click **Edit**, set **Hidden** to **Yes**, and then click **Update**.

Hiding a document in either of these ways sets the **Obsolete** flag on the **Document** entity and does not retire the document in the database. You can view hidden documents by setting **Include Hidden Documents** to **Yes** in the search section of the **Documents** screen.

Hiding a document is not the same as deleting it. The **docdelete** permission is necessary to delete documents. Only users who have that permission can delete documents.

Delete a document

Procedure

1. Open the **Documents** screen and select the document in the **Documents** list.
2. Click **Delete Selected**.

If this button is dimmed or there is no Delete  action visible in the **Actions** column, you might not have the authority to delete that file.

See also

- “Hiding a document” on page 684

Configuring and integrating document management

The base configuration provides Document Management system permissions, configuration parameters, plugins, and document templates that you can configure or manage as an administrator.

Guidewire recommends integrating with an external document management system rather than using the default demonstration document management system on the PolicyCenter server. The default system is useful only for demonstration purposes and does not support features of a real document management system, such as document versioning.

See also

- *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 security types for documents and assign permissions to users that relate to these security types.

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.

The following system permissions provide security for documents.

Name	Purpose of permission
<code>doccreate</code>	Create documents for a policy, account, policy transaction, or contingency
<code>docdelete</code>	Remove documents from a policy, account, policy transaction, or contingency
<code>docedit</code>	Edit documents on a policy, account, policy transaction, or contingency
<code>docmodifyall</code>	Modify any document, regardless of security type
<code>docview</code>	View documents on a policy, account, policy transaction, or contingency
<code>docviewall</code>	View any document, regardless of its security type

See also

- “Access control for documents and notes” on page 717

Configuration parameters for document management

The following configuration parameters in the `config.xml` file control search for and the display and editing of files in a document management system.

- `DisplayDocumentEditUploadButtons`
- `DocumentContentDispositionMode`
- `DocumentTemplateDescriptorXSDLocation`
- `FinalDocumentsNotEditable`
- `MaximumFileUploadCount`
- `MaximumFileUploadSize`
- `MaximumTotalUploadSize`

Another section of the `config.xml` file maps document file types—also called MIME types—to file extensions and associated icons in the user interface. For example:

```
<mimetypesmapping>
  <mimetype name="application/msword"
    extensions=".doc"
```

```
icon="mime_word_16.png"
<!-- more mappings -->
</mimetypemapping>
```

See also

- To configure search parameters for documents, see “Searching for documents” on page 680.
- For details about document management and related integration points, see 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 the plugin implementation class registered in this plugin registry. The class searches its metadata and returns a list of documents found.</p> <p>You can implement your own plugin implementation class to interface with a system for storing document metadata—name, id, status, author, and so on. If the plugin is not enabled, 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 plugin is disabled, and the following plugin implementation class is registered:</p> <p><code>gw.plugin.document.impl.LocalDocumentMetadataSource</code></p>
IDocumentContentSource	<p>PolicyCenter passes to the plugin implementation class registered in this plugin registry the metadata for one document. The registered class registered returns the document content and does the following:</p> <ul style="list-style-type: none"> • Interfaces with a document storage system. • Contains methods for creating, updating, and retrieving document contents. • Supports the following document retrieval modes: <ul style="list-style-type: none"> ◦ Document contents. ◦ Gosu executed by client rules. ◦ URL to a server content store. <p>In the base configuration, the following plugin implementation class is registered:</p> <p><code>gw.plugin.document.impl.AsyncDocumentContentSource</code></p> <ul style="list-style-type: none"> • In the registry, the parameter TrySynchedAddFirst is set to true and SynchedContentSource is set to <code>gw.plugin.document.impl.LocalDocumentContentSource</code>. • These parameter values cause the class to first try to use synchronous document management. If it fails, then it uses asynchronous document management.
IDocumentProduction	<p>This plugin registry registers a plugin implementation class that is the interface to a document creation system.</p> <p>The document creation process can:</p> <ul style="list-style-type: none"> • Involve extended workflow or asynchronous processes or both. • Depend on or set document fields. <p>In the base configuration, the following plugin implementation class is registered:</p> <p><code>gw.plugin.document.impl.LocalDocumentProductionDispatcher</code></p>
IDocumentTemplateSource	<p>This plugin registry registers a plugin implementation class that searches for and retrieves templates describing the document to be created. In the base configuration, the plugin implementation class is:</p> <p><code>gw.plugin.document.impl.LocalDocumentTemplateSource</code></p>

Interface	Description
IDocumentTemplateDescriptor	This interface describes the templates used to create documents. It include basic metadata (name, MIME type, and so on) and a pointer to the template content. In the base configuration, a class that implements this interface is: <code>gw.plugin.document.impl.XMLDocumentTemplateDescriptor</code>

See also

- *Integration Guide*

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 edit copies of these examples. The descriptor file is in XML format. Studio does not provide a special editor to help generate new templates.

See also

- For details about document management, document templates, and related integration points, see the *Integration Guide*.
- To automatically create documents by using rules, see the *Integration Guide*. Use similar rules to create a document in a workflow.

Smart Communications for PolicyCenter

Smart Communications for PolicyCenter integrates with SmartCOMM and enables you to create and edit documents in the PolicyCenter user interface and in bulk using SmartCOMM templates. In the base configuration, the integration produces documents in PDF format. Through configuration, you can add support for other channels such as HTML and email. This integration requires that you have Smart Communications SmartCOMM product. The integration does not include any SmartCOMM templates. You must create and configure templates to work with the integration. You must purchase and license SmartCOMM separately. The integration uses a subset of SmartCOMM features.

The integration uses SmartCOMM for the following functionality:

Document production

SmartCOMM provides template design and the ability to create documents using SmartCOMM templates. In the base configuration, the integration produces documents in PDF format. Through configuration, you can add support for other channels such as HTML and email.

Asynchronous bulk document production

Asynchronous bulk document production in Smart Communications for PolicyCenter uses batch processing in PolicyCenter to create documents in bulk. The SmartCOMM Appliance creates the documents and generates PDF files. PolicyCenter and SmartCOMM use a shared NFS folder to exchange data and documents.

Synchronous document production

Create on-demand documents from the PolicyCenter user interface automatically or at the user's request. For example, when you select **Quote**→**Print Quote**, PolicyCenter automatically creates the policy document. When the user selects a SmartCOMM template from **New Document** on the **Documents** screen of a policy transaction, PolicyCenter creates a document using the selected template.

SmartCOMM browser-based draft editor

The integration uses the SmartCOMM browser-based draft editor to display and edit documents in PolicyCenter on the **Draft Editor** panel.

Third-party software requirements for Smart Communications for PolicyCenter

Smart Communications for PolicyCenter requires the following third-party software which you must purchase and license separately:

- **SmartCOMM** – A multi-tenant cloud system. Tenancy is required to design, edit, and create documents.
- **SmartCOMM Appliance**– A virtual machine for bulk production and interactive on-demand document creation that is installed in the same local or virtual network as the PolicyCenter servers. The appliance contains its own operating system, application, and server. Also referred to as SC Appliance.

Smart Communications for PolicyCenter was built and tested against SmartCOMM API, version 4.

Note: During development of Smart Communications for PolicyCenter, the Smart Communications division of Thunderhead was launched as an independent company. References to Thunderhead may still exist in features and configuration files.

Special features in Smart Communications for PolicyCenter

Smart Communications for PolicyCenter is an integration with a document production system, but the implementation has some differences from what is described in .

When you enable the integration, Smart Communications handles all document production. In PolicyCenter, you can only choose Smart Communications templates when creating a new document in the **Documents** panel.

In general, document production integrations provide the ability to produce documents in a single format, such as Microsoft Word. Smart Communications provides the ability to produce documents in multiple formats, including PDF and HTML, among others. Therefore, this integration has its own implementation of the **IDocumentProduction** plugin.

With other document production integrations, you cannot view the document in the PolicyCenter user interface. This integration uses the SmartCOMM browser-based draft editor to provide viewing and editing of the document in the PolicyCenter user interface. Therefore, this integration includes a **DraftDocumentHandler** plugin which enables the SmartCOMM browser-based draft editor.

Document types in Smart Communications for PolicyCenter

Smart Communications for PolicyCenter enables you to create editable documents and completed documents based on SmartCOMM templates in PolicyCenter. You can view documents and modify editable documents in PolicyCenter.

Completed documents

The information in *completed documents* is static, meaning that the content does not vary. You cannot edit the content or view completed documents in the SmartCOMM draft editor in PolicyCenter. In the **Documents** panel, the **Status** of complete documents is always **Final**. In the SmartCOMM template for completed documents, you can provide boilerplate text as well as placeholders for text that will be retrieved from PolicyCenter objects.

Examples of completed documents are:

- Declarations page describing terms and conditions for a policy

If you are integrated with a content management system, you can implement code to upload completed documents to that system.

Editable documents

Editable documents contain information that is not final and that you can edit. In PolicyCenter, you can edit a draft document multiple times and save your changes to the draft. When you are finished editing, you can create the document, changing it to a completed document. The completed document is no longer editable.

Editable documents are in **Draft** status and are stored in XML format in the PolicyCenter database.

Network flows in Smart Communications for PolicyCenter

With Smart Communications for PolicyCenter, PolicyCenter interacts with the following external systems:

SmartCOMM Appliance

A virtual machine for bulk production and interactive on-demand document creation that is installed in the same local or virtual network as the PolicyCenter servers. The appliance contains its own operating system, application, and server. Also referred to as SC Appliance.

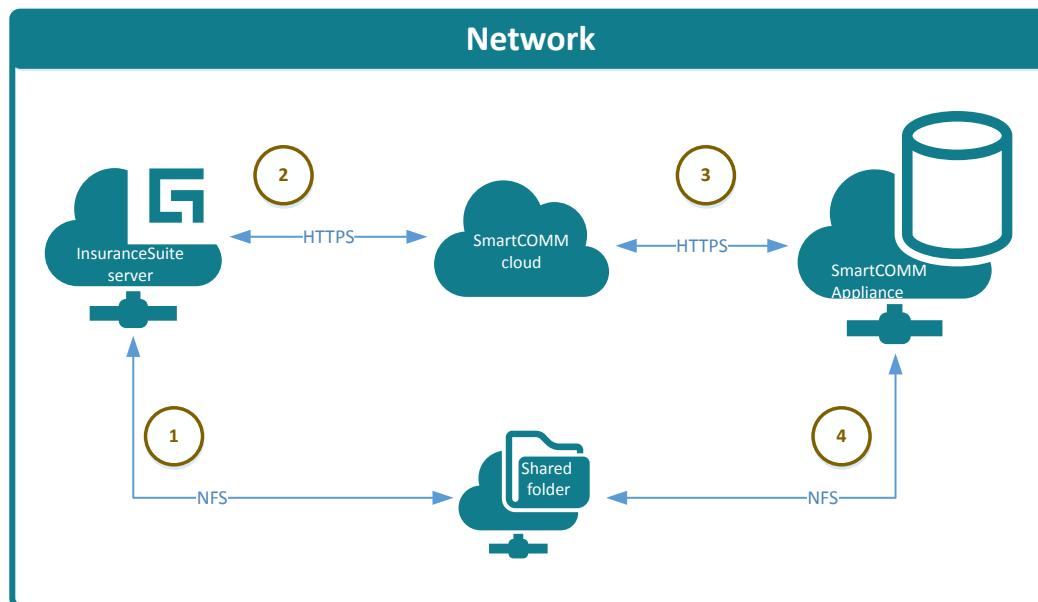
SmartCOMM Cloud

Provides communication through HTTPS with SmartCOMM.

Shared folder

PolicyCenter and SC Appliance exchange files through a temporary data store accessible through NFS.

The following illustration shows the network flow:



Smart Communications for PolicyCenter provides the ability to produce documents from SmartCOMM templates asynchronously at the user's request or in bulk using batch processing. In the base configuration, the flow for both types of document generation is as follows:

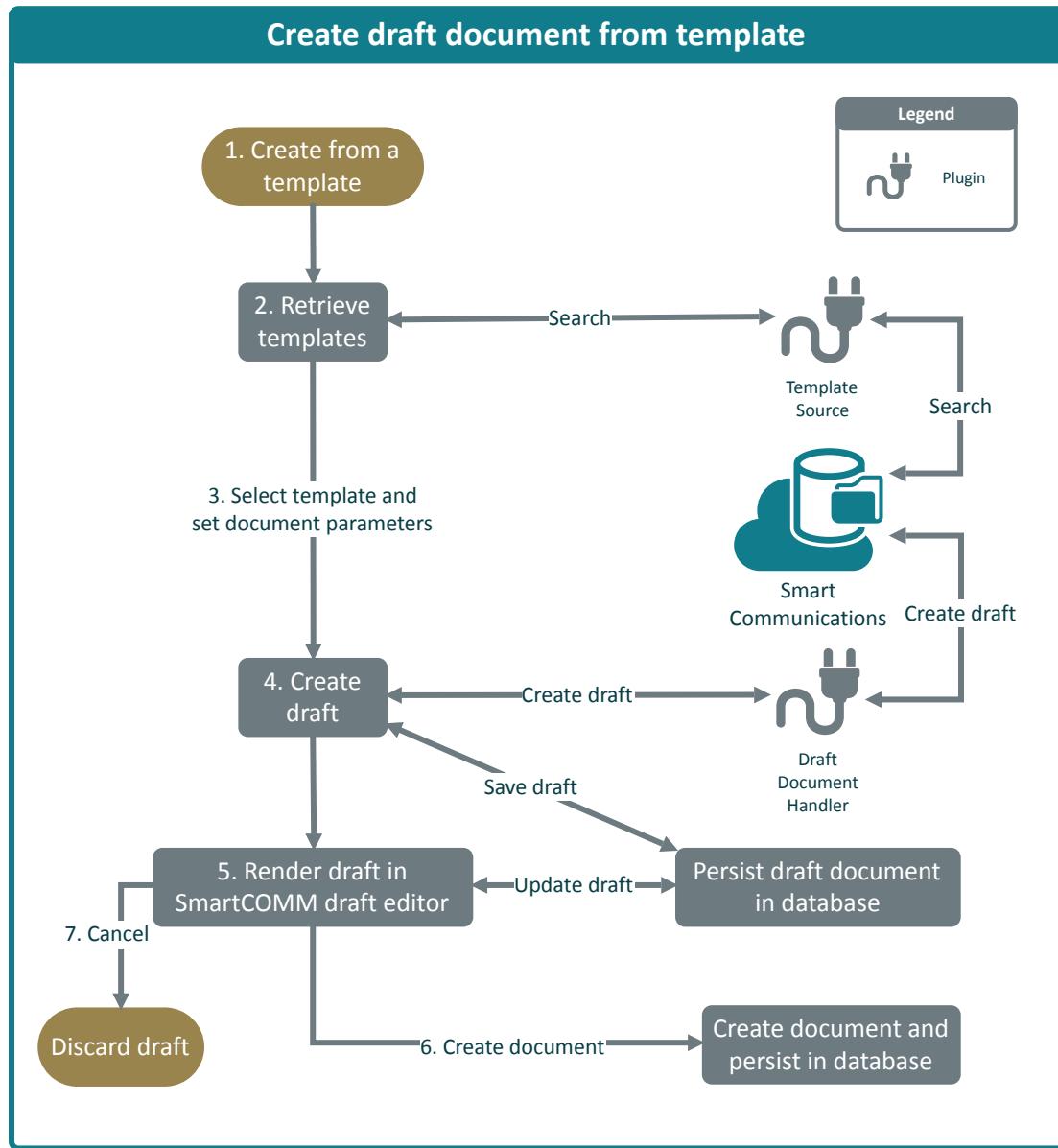
1. For every document that is created, PolicyCenter writes XML through NFS to the shared folder. The XML is in Guidewire XML format and consists of the document contents and other information.
2. PolicyCenter sends a message through HTTP alerting SmartCOMM that new documents are waiting to be processed.
3. SmartCOMM connects to the bulk appliance which processes the documents and generates PDF files.
4. SmartCOMM writes the PDF files through NFS to the shared folder.

Synchronous draft document creation flow in Smart Communications for PolicyCenter

When creating a document synchronously, you can create a new draft document by selecting **New Document**→**Create from a template** from the **Documents** screen.

Application flow

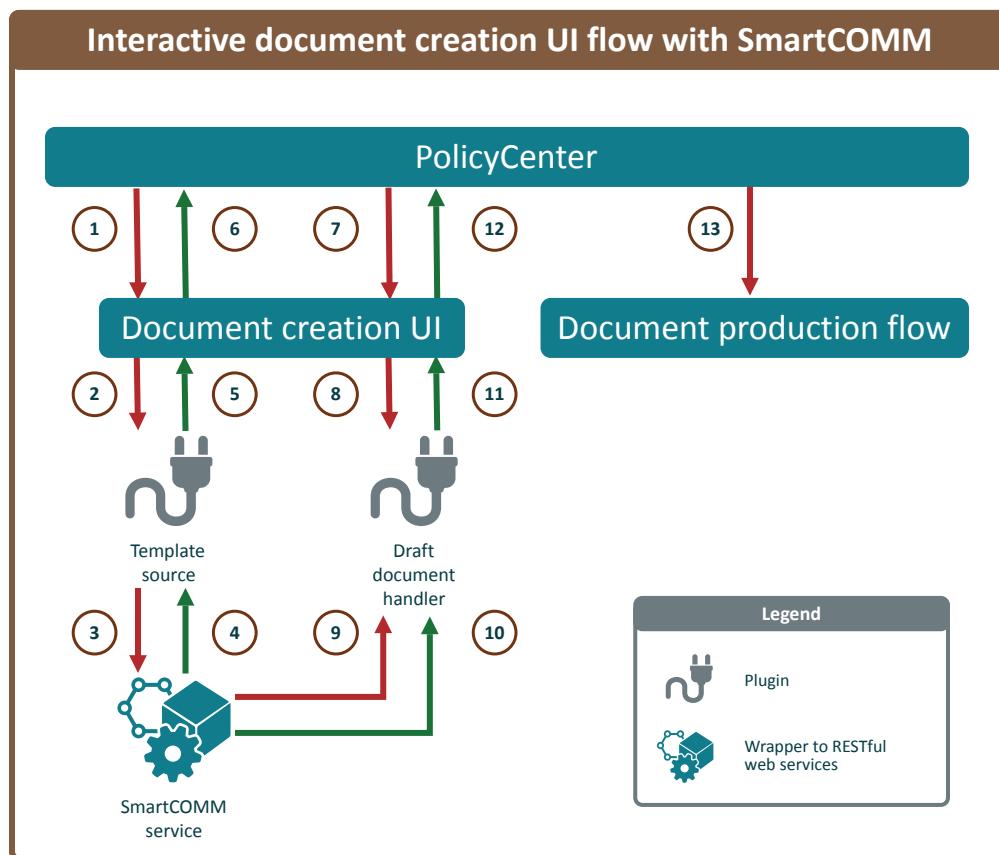
At the application level, creating a new draft document triggers the following flow:



1. In PolicyCenter, you select **Create from a template**.
2. The document creation user interface, defined in the PCF file, calls the Template source plugin implementation to retrieve the templates from SmartCOMM.
3. The user selects a template from the **Search Results** of the **Upload Documents** tab and then defines properties such as the name of the document.
4. The user clicks **Create Draft**. The Draft Document Handler plugin calls SmartCOMM to create a draft document.
5. PolicyCenter displays the SmartCOMM draft editor login screen. After you enter your credentials, the draft editor displays the draft document in the **Upload Documents** tab. You can edit the document in PolicyCenter. PolicyCenter displays a message if the template contains no fields for editable content.
6. If you click **Create Document**, PolicyCenter calls the document production flow, which creates the **Document** object.
7. If you click **Cancel** instead, then PolicyCenter deletes the draft document and does not create the **Document** object.

Integration flow

At the integration level, creating a new draft document triggers the following flow:



In the illustration, the following actions occur at each numbered item:

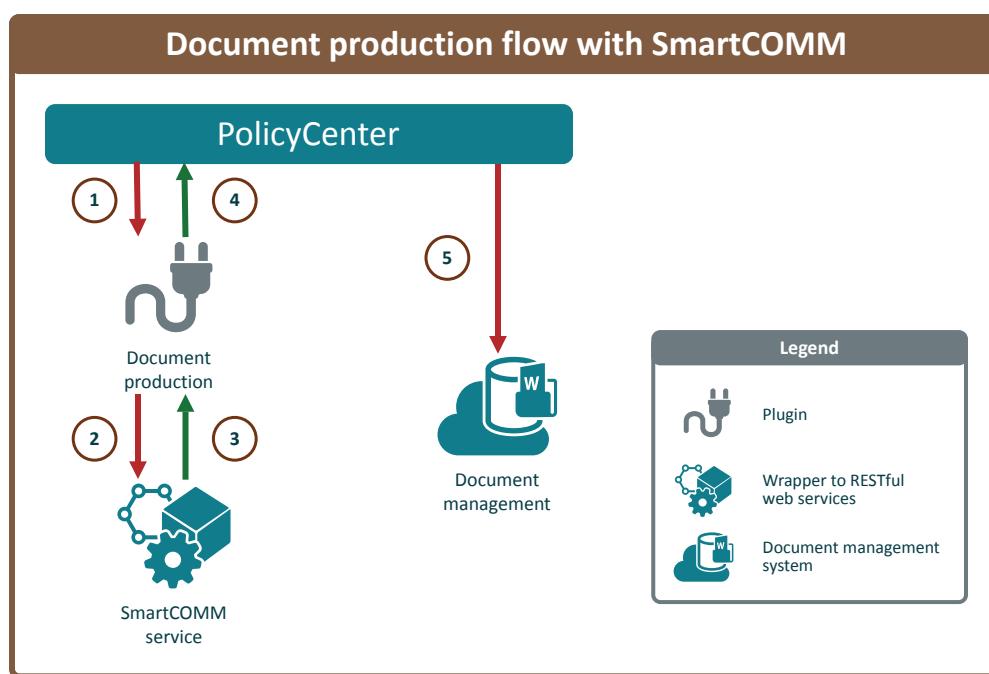
- In PolicyCenter, you select **Create from a template**. (Step 1)
- The document creation user interface, defined in the PCF file, calls `SmartCommsDocumentTemplateSource`, the `IDocumentTemplateSource` plugin implementation. The plugin accesses the `SmartCommsConfiguration` entity instance. The plugin parameters specify:
 - The effective date of the template.
 - Maximum number of results.
 - Keywords, language, and document type for filtering the templates. This filter information comes from the PCF file.
- The `SmartCommsDocumentTemplateSource` plugin requests the templates from SmartCOMM. (Step 3)
- SmartCOMM retrieves the templates and provides them to the `SmartCommsDocumentTemplateSource` plugin. (Step 4)
- The `SmartCommsDocumentTemplateSource` plugin returns the list of templates to the user interface, which displays these templates in the **Search Results** of the **Upload Documents** tab. (Step 5)
- The user interface returns the template descriptors to PolicyCenter, which displays these templates in the **Search Results** of the **Upload Documents** tab. (Step 6)
- You select a template, and then define properties such as the name of the document. You click **Create Draft**. PolicyCenter stores this information in transient objects. PolicyCenter does not yet create the `Document` entity instance in the database. (Step 7)
- The user interface calls the `DraftDocumentHandler` plugin. (Step 8)
- The `DraftDocumentHandler` plugin sends a message to SmartCOMM through RESTful web services to create a draft document. (Step 9)
- SmartCOMM returns the draft document to the `DraftDocumentHandler` plugin. (Step 10)

11. The `DraftDocumentHandler` plugin returns the draft document to the user interface. PolicyCenter displays the SmartCOMM draft editor login screen. After you enter your credentials, the draft editor displays the draft document in the **Upload Documents** tab. PolicyCenter displays a message if the template contains no fields for editable content.
12. You edit the document in PolicyCenter.
13. If you click **Create Document**, PolicyCenter calls the document production flow, which creates the **Document** object. If you click **Cancel** instead, then PolicyCenter deletes the draft document and does not create the **Document** object.

In the base integration, you can create documents up to 260,000 characters. You can adjust this limit in Content property of the `DraftDocument` entity.

Document production flow with Smart Communications for PolicyCenter

PolicyCenter starts the document production flow after it finishes creating the draft document. If successful, the document production flow results in a **Document** object, which holds the completed document. The document production flow triggers the following process:



In the illustration, the following actions occur at each numbered item:

1. PolicyCenter, triggered either by the synchronous or asynchronous document creation flow, calls `SmartCommsDocumentProduction` plugin. This plugin implements the `IDocumentProduction` plugin interface.
2. The `SmartCommsDocumentProduction` plugin calls SmartCOMM to create the document from the draft document.
3. SmartCOMM generates the document and returns it either as an input stream for synchronous production or as an XML payload for asynchronous production.
4. The `SmartCommsDocumentProduction` plugin returns the document to PolicyCenter.
5. PolicyCenter creates a **Document** object and saves it in the PolicyCenter database or in the document management system, if one is integrated.

See also

- “Synchronous draft document creation flow in Smart Communications for PolicyCenter” on page 691

Asynchronous bulk document production in Smart Communications for PolicyCenter

Asynchronous bulk document production in Smart Communications for PolicyCenter uses batch processing in PolicyCenter to create documents in bulk. The SmartCOMM Appliance creates the documents and generates PDF files. PolicyCenter and SmartCOMM use a shared NFS folder to exchange data and documents.

The integration performs the following actions:

- Retrieves SmartCOMM templates.
- Sends XML payload in GX model format to SmartCOMM. SmartCOMM generate a single document or documents in bulk from the payload.
- Creates Document entity instances.
- Calls the document production flow.

Attaching documents to objects in Smart Communications for PolicyCenter

In Smart Communications for PolicyCenter, the integration attaches documents to PolicyCenter objects. In the PolicyCenter base configuration, you can attach documents to the following objects:

- Account
- Policy
- Job – Referred to as policy transaction in the user interface
- Contingency

Documents created in activities

Although activities do not contain documents, on the **Activity Detail** tab, you can select **Create Document From Template**. The **Activity** entity does not have a **Documents** array. Documents that you create on an activity are associated with the entity with which the activity is associated. The following table shows which entity contains the document associated with the activity:

Add document to activity in	Entity associated with document
Submission policy transaction	Policy
Account	Account
Other policy transactions	Job
Contingency	Policy which contains the contingency

Specifying when Smart Communications for PolicyCenter creates documents

Smart Communications for PolicyCenter enables you to specify when PolicyCenter creates, or binds, documents from SmartCOMM templates. The integration defines three values that you can specify in the

DocumentProductionBindingType parameter in config.xml. From earliest to latest binding, the parameter values are:

- EARLY – With early binding, creating, modifying, or other actions on a document causes document production to generate the document contents.
- EARLY_DEFERRED – Early deferred binding occurs a bit later in the process than early binding. With early deferred, document production generates the document contents after establishing communication with the SmartCOMM messaging infrastructure.
- LATE – With late binding, Bulk Submission Job batch processing generates the document contents.

The base configuration is configured for early binding. Through configuration, you can implement early deferred and late binding.

See also

- *Integration Guide*

PolicyCenter administration

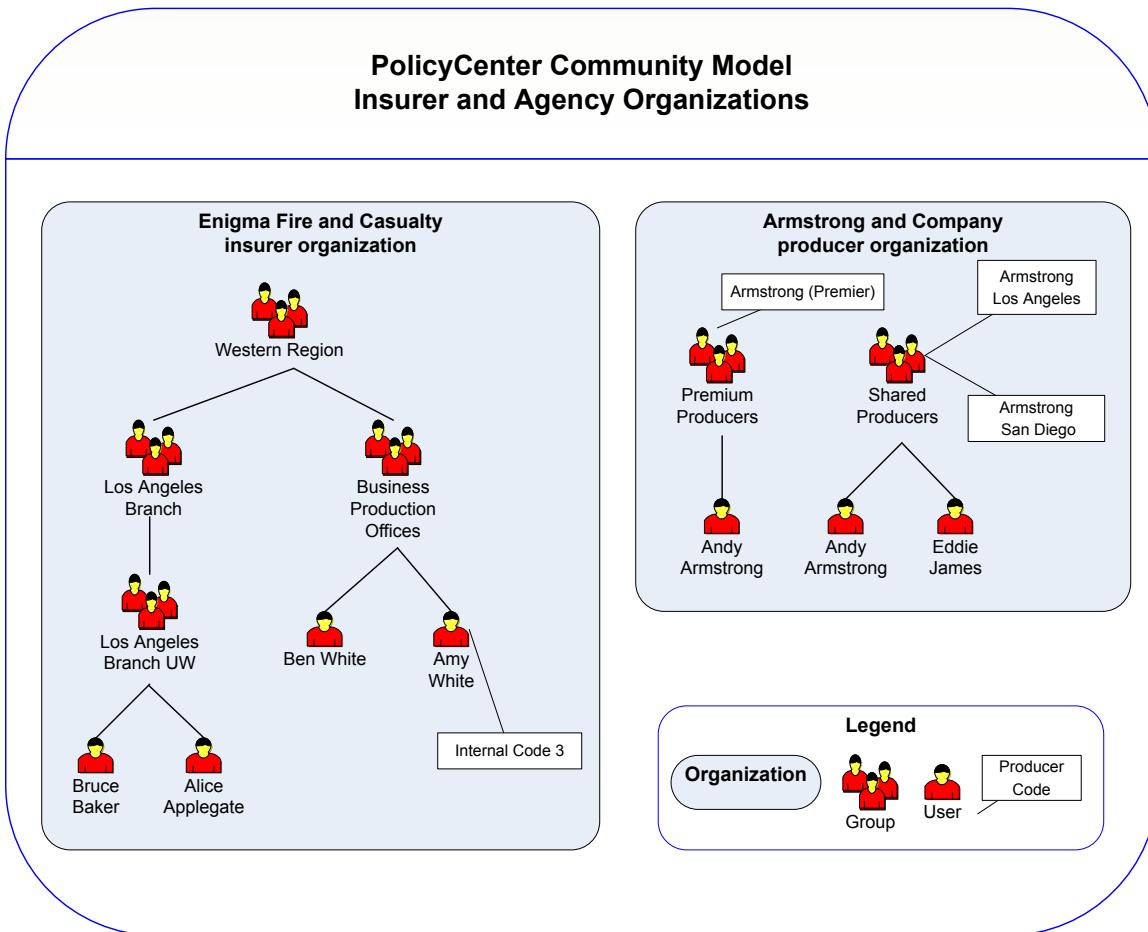
Security: roles, permissions, and the community model

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 producer organizations. To perform these tasks, a user must be assigned a role with the appropriate permissions. The sample data in the base configuration includes the Community Admin role. A user with this role can work with the community model. This user can assign permissions to 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 producer organizations.

PolicyCenter also allows you to control access to documents and notes through access permissions.

Community model overview

PolicyCenter uses a *community model* to organize users into producer organizations. The organization of the insurer 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 insurer organization and the Armstrong and Company agency organization.



There are two types of users:

- **Internal** – An internal user is an employee of the insurer. An internal user can potentially see groups and permissions of external users. In the illustration, the users in the Enigma Fire and Casualty insurer organization are internal users.
- **External** – An external user is a person outside the insurer 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 insurers. 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 insurers.

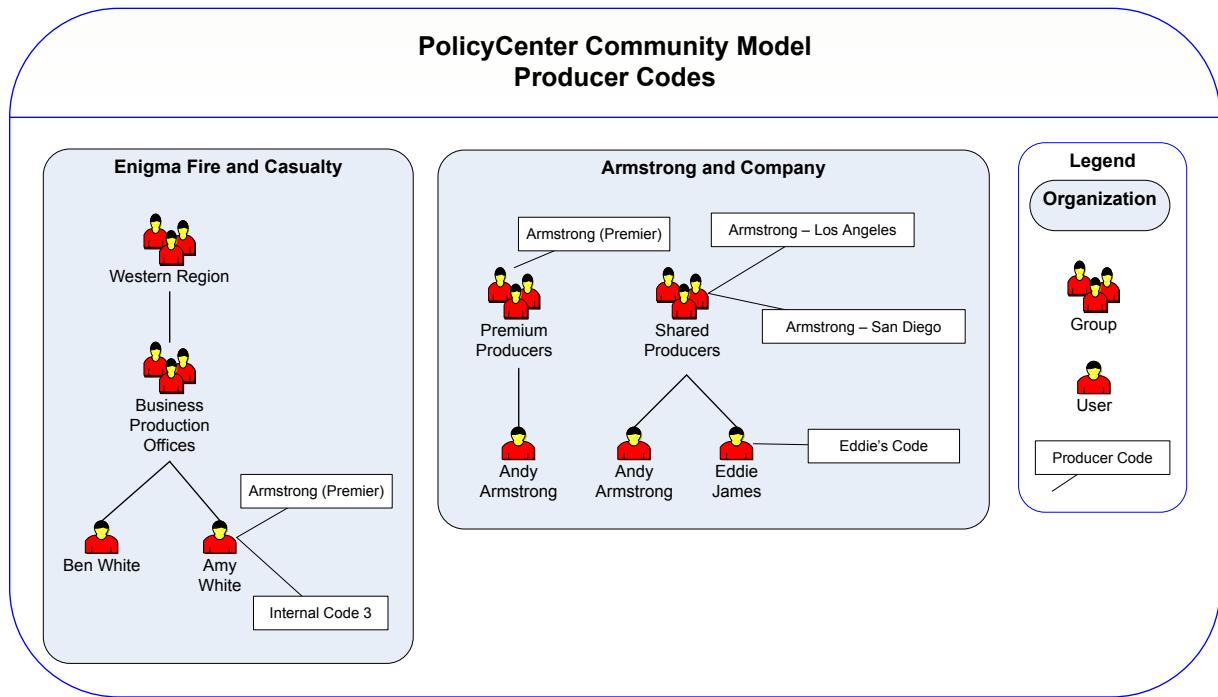
Within an organization, users can be assigned to one or more groups. In the illustration above, the user, Andy Armstrong, belongs to the Premium Producers and the Shared Producers groups.

Producers in the community model

A *producer code* is a unique identifier assigned by the insurer 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 producer 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 insurer 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.



Insurers 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.

Types of security

Role-based security

Grants permission to perform actions such as create a submission or edit an account.

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 Rule Admin role 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

Defines a user's access to data.

Data-based security for accounts and policies defines exactly which accounts and policies you have access to.

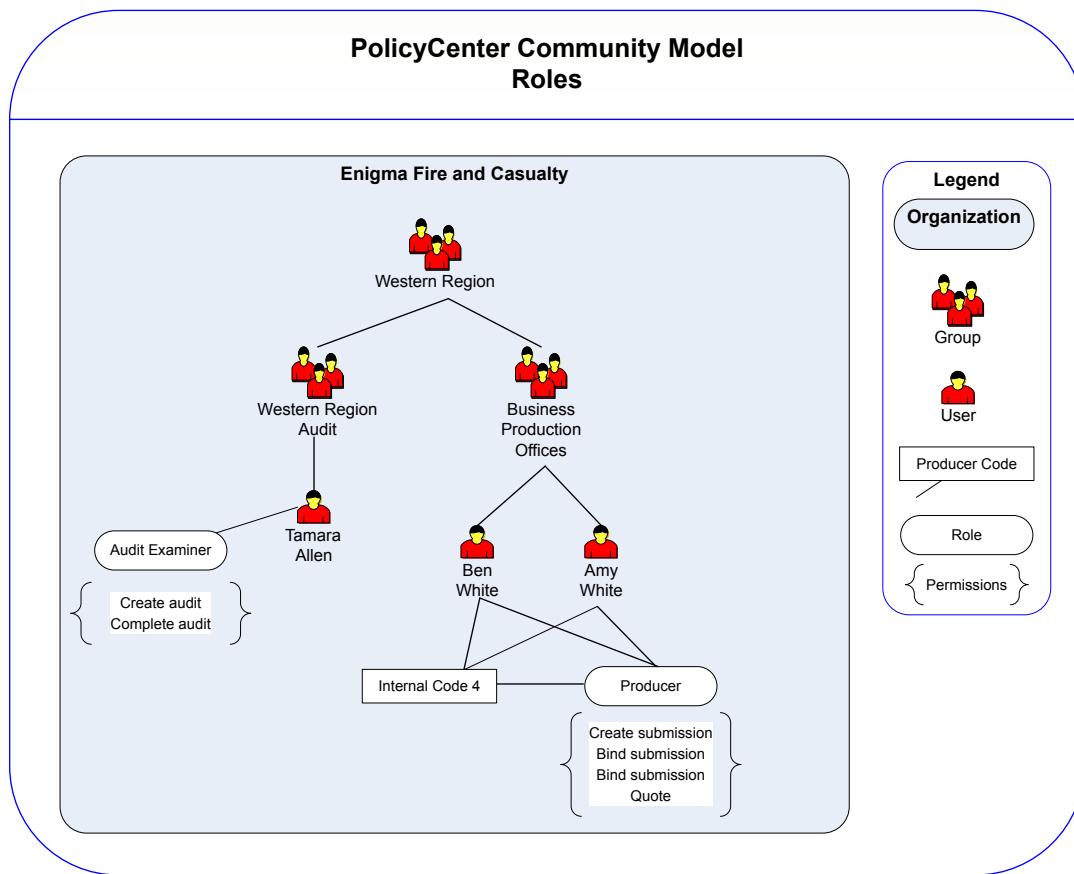
PolicyCenter manages this security through producer codes and producer organizations. An insurer assigns producer codes to producers. Producers can be internal or external. You can think of producer codes as enforcing the agreement between the producer and the insurer that determines exactly what a producer can do on an account or policy. An insurer can attach permissions to producer codes, then attach producer codes to users and groups. This arrangement allows an insurer 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 701 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

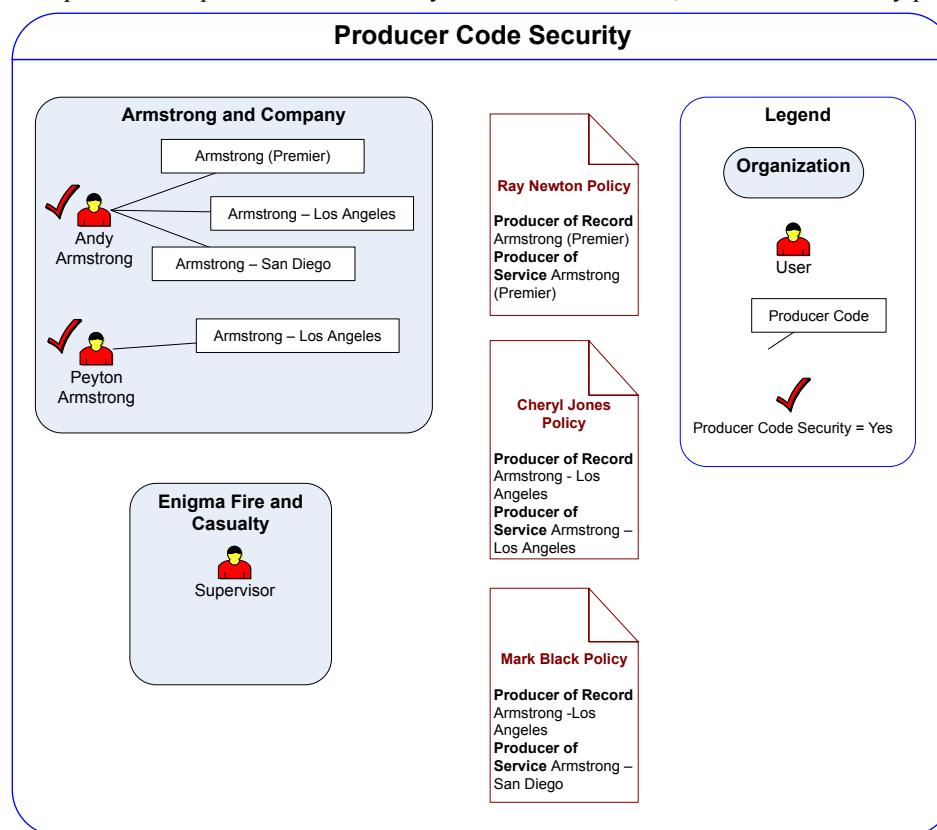
Producer code *roles* allow you to create groups of permissions and attach them to producer codes. You can then assign the producer codes to users, to policies, or accounts. Users who have producer code security can only use their role permissions against policies that have a producer code matching one 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 “Turn on producer code security” on page 711.

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 707.

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 (producer organizations, groups, users, and other administrative data).
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 insurer’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.
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 insurer’s underwriting supervisor. Also includes additional management permissions.

Role name	Role type	Description
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 users and security” on page 709.

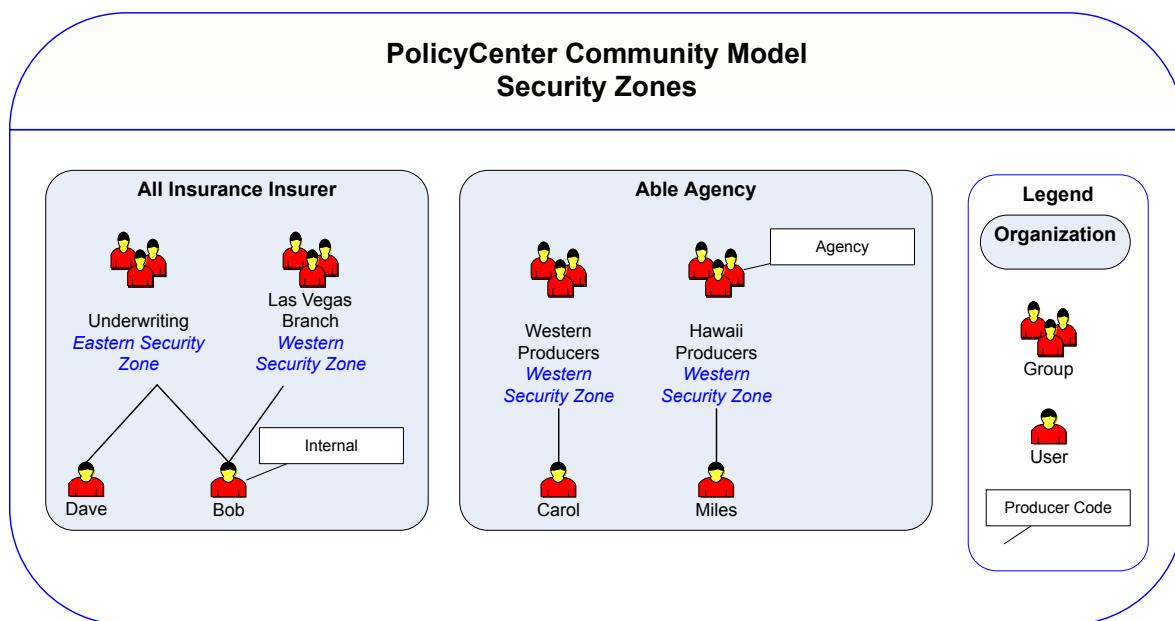
Account-level security

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 comprises all producer codes of service from bound policies on the account. You can modify this logic with Gosu in Studio in the `ensureProducerOfServiceFromAccount` method in `PolicyPeriodBaseEnhancement.gsx`.

Data-based security for the community model

Limits access to users, groups, producer organizations, and producer codes.

PolicyCenter also has *data-based security for the community model* that limits the current user’s access to users, groups, producer 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 Insurer 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 Insurer 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

Fine tune user access.

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 producer 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 insurer, 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, producer 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 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 a producer *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.

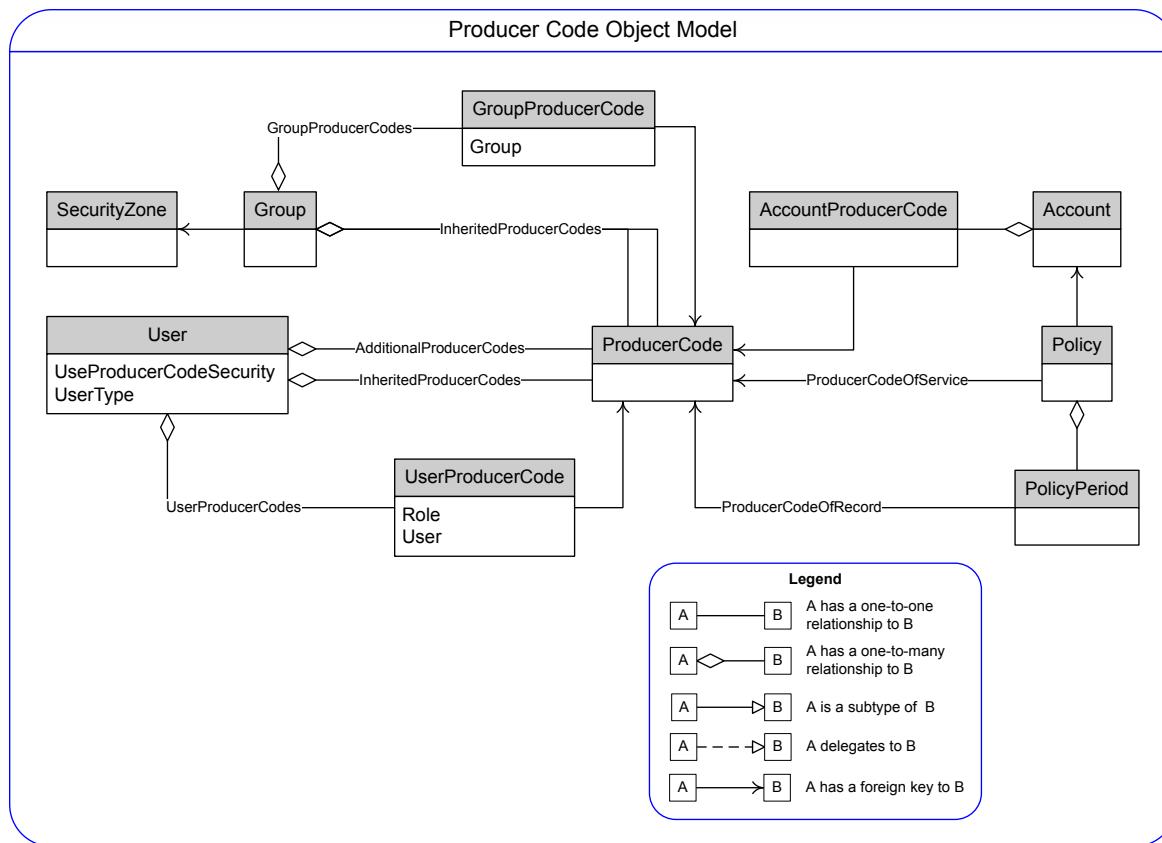
To create an external user, set the **Internal User** option set to **No** on the **Basic** tab in the **User** screen. When you create an external user, the fields and available values in the **User** tabs change. Required fields appear with an asterisk: *.
Unlike internal users, external users must 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 users and security

To control the level of access to PolicyCenter data, you manage it primarily from the **Administration→Users & Security** menu in the user interface. To a lesser extent, you also control access in plugins.

View permissions on selected roles

About this task

To see or edit the list of permissions on selected roles:

Procedure

1. Navigate to **Administration**→**Users & Security**→**Roles**.
2. Click the link in the **Name** column to view a role.

Create a permission

Procedure

1. In Studio, navigate to **configuration**→**config**→**Extensions**→**TypeList** and open **SystemPermissionType.ttx**.
2. Right-click in the **TypeList** table and select **Add new**→**typecode**. 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`

Remove a permission

Procedure

1. In Studio, navigate to **configuration**→**config**→**Extensions**→**TypeList** and open **SystemPermissionType.ttx**.
2. Right-click the typecode and select **Remove**. If **Remove** is not available, you can select the typecode for the permission, and change **retired** to **true**.
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.

Add a permission to a role

About this task

To add a permission to a role:

Procedure

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.

Add a new role

Procedure

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**.

Remove a role

Procedure

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.

Turn on producer code security

About this task

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.

See also

- “Producer code security and policies” on page 703

Procedure

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**.

Working with regions

In PolicyCenter, you can group users into units that may map to geographical regions, divisions, or departments. PolicyCenter assigns activities and other assignable objects to users and groups. (See the *Rules Guide* for more information.)

In the base configuration data model, **Region** has a property to access its **Organization**. You can use **Organization** to represent an external agency. You can write assignment rules that route policies to the appropriate processor or underwriter depending upon the region associated with the agency.

To view regions, navigate to the **Administration**→**Users & Security**→**Regions** screen. This screen displays the first page of regions. In the base configuration, if there are more regions than fit onto one page, you can search to filter the list of regions.

The **SearchOn** property controls the visibility of search fields on the **Regions** screen. In the base configuration, the values for the **SearchOn** property are:

- **ALWAYS** – Always display search fields on the **Regions** screen.
- **NEVER** – Never display search fields on the **Regions** screen.
- **AUTO** – Display search fields if the number of regions is two or more pages. This is true if the number of regions is greater than **ListPageSizeDefault**.

The **SearchOn** property is defined in **RegionsUIHelper.gs** in the **gw.admin** package. In the base configuration, this property is set to **AUTO**.

The **ListPageSizeDefault** parameter in **config.xml** defines the number of rows to display in a list view.

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
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 the *Guidewire Contact Management Guide*.

Working with affinity groups

Some insurers write policies based on the insured belonging to a certain group. For these insurers, the lines of business in the base configuration enable you to associate an affinity group with a policy. Affinity groups are defined in the **Administration** tab.

The **Affinity Group Administration** (`affinitygroupadmin`) permission controls whether a user can administer affinity groups. In the default configuration, the **Community Admin** and **Underwriting Supervisor** have this permission.

See also

- “Affinity groups” on page 166

Creating a new affinity group

To create a new affinity group, navigate to the **Administration** tab and select **New Affinity Group** from the **Actions** menu. This command appears only if you have `affinitygroupadmin` permission. Selecting this command displays the **New Affinity Group** screen where you can define a new affinity group.

The **New Affinity Group** screen in the **Administration** tab (`NewAffinityGroup.pcf`) displays four tabs:

- **Basics** – Basic information about the affinity group including name, type, producer organization, contact, and start/end dates. Leave the dates blank to enable the affinity group to be applied to any policy effective date.
- **Jurisdictions** – The states in which the affinity group can be applied. Leave blank to enable the affinity group to be applied to all states.
- **Products** – The lines of business in which the affinity group can be applied. Leave blank to enable the affinity group to be applied in all lines of business.
- **Producer Codes** – The producer codes for which the affinity group can be applied. Leave blank if you are defining an open affinity group that you want any producer to be able to use.

The **Name** and **Type (Open or Closed)** are required fields. Additionally, if you set the type to **Closed**, validation requires that you specify an **Organization** and at least one **Producer Code**. Each additional detail that you include in an affinity group definition further restricts its availability accordingly. For example, if you specify a start and end date, the affinity group is unavailable to a user writing a policy with an effective date outside this range of dates.

Searching for affinity groups

An **Affinity Groups** command appears in the left sidebar of the **Administration** tab. This command appears only if you have `affinitygroupadmin` permission. Selecting this command displays the **Affinity Groups** search screen where you can search for affinity groups and view or edit those that are found.

The **Affinity Groups** search screen (`SearchAffinityGroup.pcf`) enables you to search by several criteria, and to search for only available groups. The **Search Only Available Groups** option restricts the search to affinity groups whose date range, if specified, includes today's date.

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 insurers 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 commercial 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 commercial 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 insurer 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 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 insurer'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 security for agents working for multiple agencies

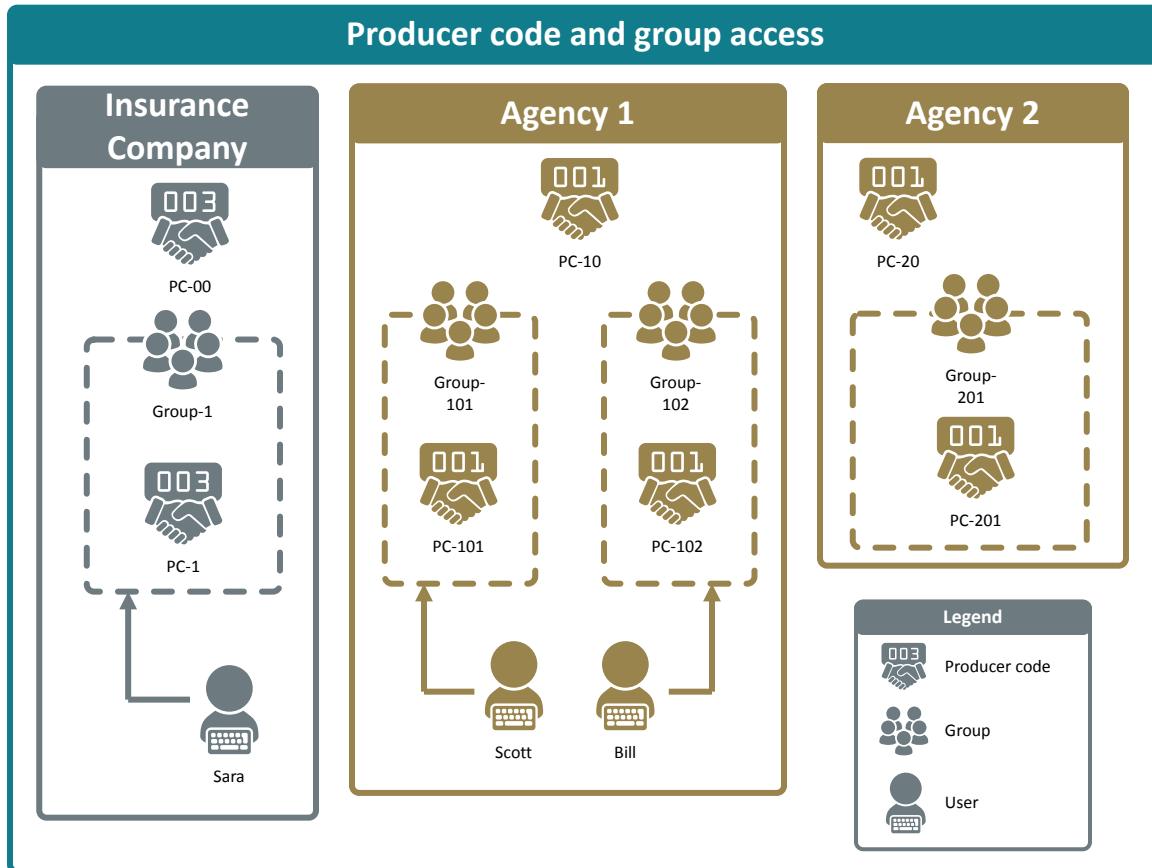
In the base configuration, PolicyCenter assumes that agents are external users working on policies for a single agency. In PolicyCenter, the agency is represented by an organization. External users can be assigned to groups only within their organization. Users inherit producer codes from their groups. In addition, external users can be assigned producer codes from their organization.

However, some agents work on policies for more than one agency (organization). In the base configuration, this agent needs a username for each organization. Through a configuration parameter, PolicyCenter enables you to assign an external user to groups and producer codes from more than one organization. The **ExternalUserAccess** configuration parameter has three settings. In order of increasing access to groups and producer codes, these settings are:

- **FULLYRESTRICTED** – Default. With Fully Restricted, external users can be assigned to groups only within their organization. Users inherit producer codes from their groups. In addition, external users can be assigned producer codes from their organization.
- **PROTECTINTERNAL** – With Protect Internal, external users can be assigned to any external groups. Users inherit producer codes from their groups. In addition, external users can be assigned producer codes from any external organization.
- **ALLOWINTERNAL** – With Allow Internal, external users can be assigned to any groups, internal or external. Users inherit producer codes from their groups. In addition, external users can be assigned producer codes from internal or external organizations.

As needed, you can change this parameter to increase access to groups and producer codes. However, be aware that decreasing access requires that you review all users, groups, and producer codes to ensure that they conform to the more restrictive parameter setting.

The following illustration shows producer codes, groups, and users in an insurance company and external agencies. User A-1 belongs to the Agency 1 organization and is assigned to the Group-101 group.



If access is Fully Restricted, external user Scott can be assigned producer codes from the group and organization to which the user belongs. As a member of Group-101, Scott has producer code PC-101. Because Scott belongs to organization Agency 1, he can be assigned producer code PC-10.

If access is Protect Internal, Scott can additionally be assigned to any external groups. So Scott can be assigned to Group-102 and Group-201. Through membership in these groups, Scott picks up producer codes PC-102 and PC-201. Scott can also be assigned any external producer codes. Therefore, Scott can be assigned producer code PC-20.

If access is Allow Internal, Scott can additionally be assigned to any internal or external group. So Scott can be assigned to Group-1, picking up producer code PC-1. Scott can also be assigned any internal producer code. So Scott can be assigned producer code PC-00.

See also

- *Configuration Guide*

Producer code currency

You can access producer codes by navigating to the **Administration → Users & Security → Producer Codes** screen. On the **Commission Plans** tab, you can choose among various commission plans from the billing system.

In a single currency system, you can specify one commission plan for each **Producer Code** in the **Commission Plan** field on the **Basics** tab.

In a multicurrency system, each commission plan can offer one or more currencies. You can select more than one commission plan. For each commission plan, specify the settlement currencies in which the producer can bind policies. For each producer code, you can associate only one plan per currency. Therefore, if you select USD in commission plan A, you cannot select USD in commission plan B.

Security Dictionary

The PolicyCenter *Security Dictionary* is web-based documentation that you can generate. Whenever you change the PolicyCenter data model, regenerate the *Security Dictionary* to view the changes.

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 and the Audit Supervisor 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, Underwriter, Underwriter Assistant, and Underwriting Supervisor can create submissions, too.

See also

- *Configuration Guide*

Access control for documents and notes

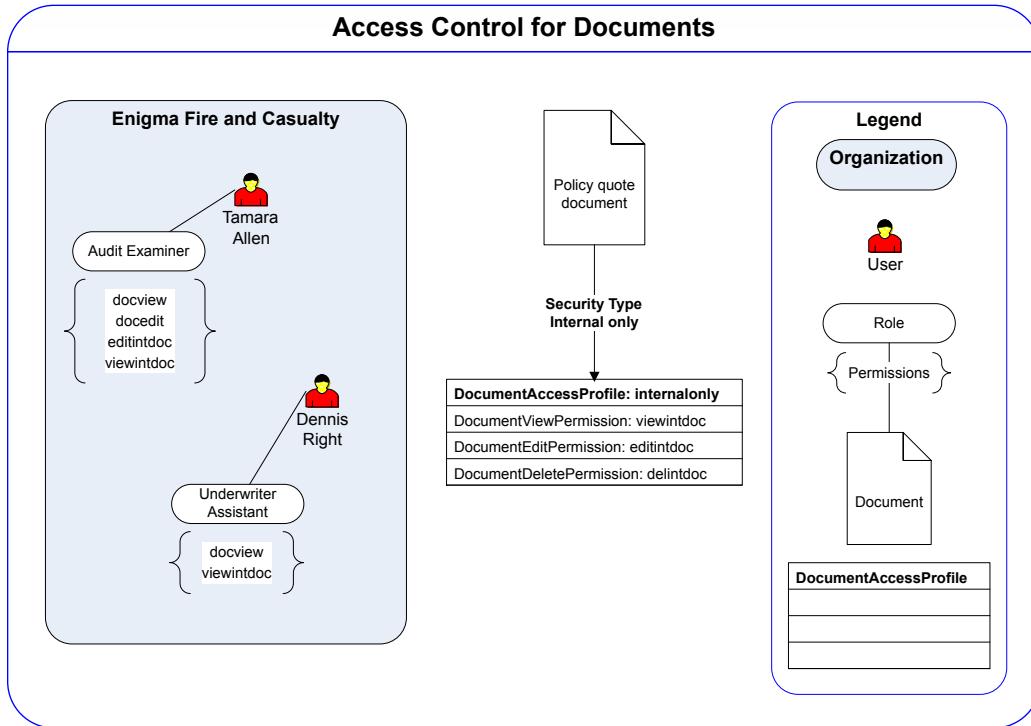
In addition to the standard document and note-related system permissions, you can control access to documents and notes by configuring access permissions. To do so, a document must have its document security type set. To see documents of a particular type, you must have both permission to view documents 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** – Security types determine document and note access control. Typical security types for documents are Unrestricted and Confidential. The supported types are defined in the **DocumentSecurityType** typelist. They appear in the **Security Type** drop-down list of the **Documents**→**Document Details** and **New Document** screens. A document can be assigned a maximum of one security type. Security types for notes are defined in a similar manner by the **NoteSecurityType** typelist.
- **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 previous two concepts, these profiles relate permissions and security types to restrict access to a subtype of documents.

In the illustration that follows, 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 topics describe how to define and use document and note access control.

Create document and note security types

A document type is set by using the **Security Type** field in the user interface or through Gosu. The security types supported in the base configuration are defined by the `DocumentSecurityType` typelist. You can add your own security types. Documents that are not assigned a security type are given the `unrestricted` security type.

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** drop-down list.

Creating 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 the following syntax, where `type` specifies a document or note security type, and `perm` is a system permission:

```

<DocumentPermissions>
  <DocumentAccessProfile securitylevel="type">      <!-- define for each security type -->
    <DocumentViewPermission permission="perm"/>    <!-- 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 the following 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>
  </DocumentAccessProfile>
</DocumentPermissions>
```

There is a similar set of permissions for notes in the `NotePermissions` element.

su, the Super User

User `su` has first name Super and last name User. In the base configuration, Guidewire configures user `su` as an unrestricted user in configuration file `config.xml`. PolicyCenter does not evaluate permissions for the unrestricted user. As a consequence of PolicyCenter bypassing permission checking, the unrestricted user has access to the entire PolicyCenter application. Thus, an unrestricted user behaves as if it has all permissions.

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.

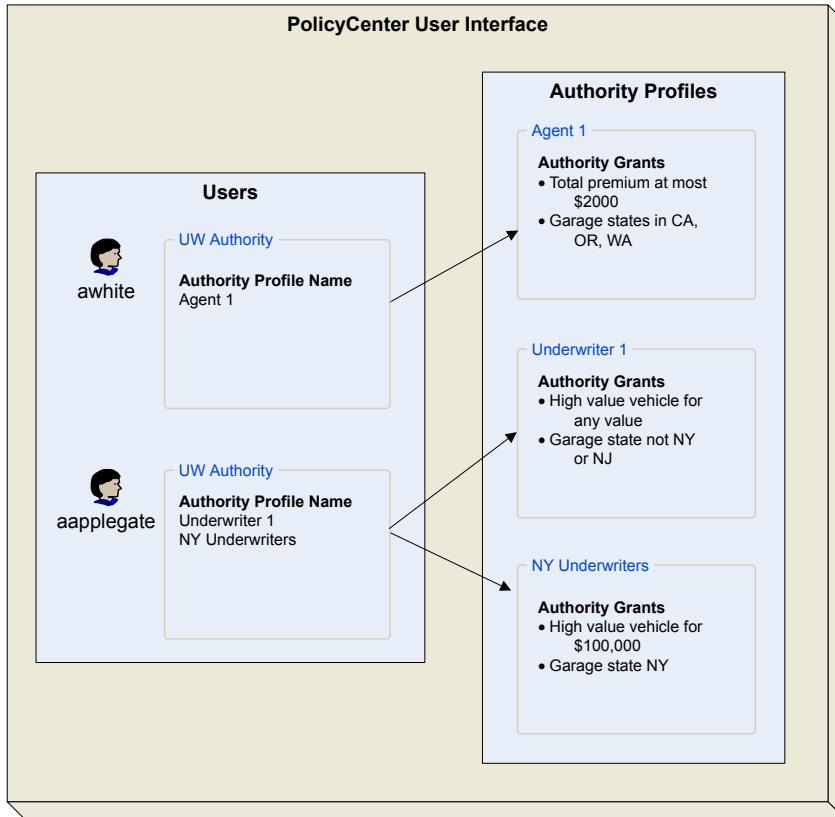
See also

- “Underwriting issues” on page 671
- *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 underwriting rules. 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 `aapplegate` 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, `aapplegate` 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.



The base application defines the samples authority profiles. You define authority profiles in the **Administration** tab of PolicyCenter. Authority profiles contain authority grants. Authority grants are associated with a particular underwriting issue type as defined by the underwriting rules.

[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.

View or edit an authority profile

About this task

Follow these steps to view or edit an authority profile.

Procedure

1. From the **Administration**→**Users & Security** menu, select **Authority Profiles**.
2. 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.
3. Click **Edit** to add or modify an authority grant.
4. Click **Add** to add an authority grant. Select a **Type**.

The authority grant uses values defined for the underwriting rules. The **Comparison** column is a value comparator. For **At most**, **At least**, and **In Set** grants, you can specify the amount or set in the **Value** 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. If so, the approval limit is increased or decreased by this amount.

Assign an authority profile to a user

About this task

You can add one or more authority profiles to a user.

Procedure

1. On the **User** screen, select the **UW Authority** tab.
2. Add authority profiles to this user.

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.

See also

- *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 the **My Groups** summary which includes:

- Your groups. Select a group to display information for that group or user.
- Links to reporting categories.
- 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, such as **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.

IMPORTANT The **Team** tab does not display system users—users with the **SystemUserType** value of **sysservices**. Refer to the *Data Dictionary* for more information on this user type.

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 on the Team tab

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

- “Assign activities on the Team tab” on page 727
- “Assign submissions, renewals, and other policy transactions” on page 727

Working with the Team tab

This topic describes how to work with the **Team** tab in the PolicyCenter user interface.

Assign activities on the Team tab

About this task

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:

Procedure

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**.

Assign submissions, renewals, and other policy transactions

About this task

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 76 and “Working with the Advanced Search tab” on page 77.

To assign a user to one or more policy transactions:

Procedure

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 an insurer 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 insurer can put a hold on issuing or changing policies in a region affected by a natural disaster.
- Regulatory changes – The insurer can put a hold on issuing or changing policies during a period of time when the insurer is finalizing insurance rates or changing coverage forms.

Policy holds overview

PolicyCenter provides policy holds for natural disasters and for regulatory changes. Policy holds are types of underwriting issues.

See also

- “Underwriting issues” on page 671

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, an insurer may decide not to bind any new policy transactions in that region before or after the natural disaster. The insurer 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 insurer 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 insurer puts an underwriting policy hold on Kauai County so that agents cannot add that coverage. The insurer 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 insurer adds those counties to the underwriting hold. Because the storm has not yet arrived, the insurer 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 insurer 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 insurer updates the hold region to Honolulu County only. Because the insurer has not assessed the full extent of damages, the insurer is not yet ready to write policies in Honolulu County. Therefore, the insurer does not set the hold end date.

Three weeks later, the insurer has assessed all damage from the hurricane. The insurer sets the hold end date to the current date. The insurer 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 “Prevent back-dating policy transaction to avoid underwriting hold” on page 731.

If, on the other hand, the hurricane completely bypassed the state of Hawaii, the insurer 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 “Policy Hold Details tab” on page 736.

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 insurer 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 insurer 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 issues” on page 671
- *Configuration Guide*

Prevent back-dating policy transaction to avoid underwriting hold

PolicyCenter prevents you from back-dating a policy transaction to avoid an underwriting hold. An insurer 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 insurer 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 insurer 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 739.

This includes topics that provide step-by-step instructions.

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 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.

Follow these steps to work through this example.

Create a policy hold

Procedure

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**→**Business Settings**→**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.

Next steps

“Create policy effective in the past” on page 732

Create policy effective in the past

Before you begin

“Create a policy hold” on page 732

About this task

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.

Procedure

1. Log in as a user with permissions to create policies. If you are using the sample data, log in as the underwriter **aapplegate**.
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 **aapplegate**.
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.

Next steps

“Renew policies through batch process” on page 733

Renew policies through batch process

Before you begin

“Create policy effective in the past” on page 732

About this task

The Policy Renewal Start batch process creates renewal policy transactions for expired or soon to be expired policies. The batch process creates a renewal policy transaction for the policy you created in the previous set of instructions.

Procedure

Run the Policy Renewal Start batch process. For instructions, see the *System Administration Guide*.

Next steps

“Approve policy hold” on page 733

Approve policy hold

Before you begin

“Renew policies through batch process” on page 733

About this task

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.

Procedure

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.

Next steps

“Issue the policy” on page 734

Issue the policy

Before you begin

Complete the step “Approve policy hold” on page 733 before you perform this step.

About this task

In the previous set of instructions, you approved the underwriting issue for the policy hold. In these instructions, you issue the policy.

Procedure

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**.

Work with policy hold actions

About this task

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 737.

The instructions assume that you have loaded the large sample data set, and that you are creating a commercial property policy.

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.

Procedure

1. As **su**, create a policy hold for commercial property policy renewals as described in “Create a policy hold” on page 732.

2. As **aapplegate**, create a commercial property submission as described in “Create policy effective in the past” on page 732.
3. Create a second commercial property policy as described in “Work with policy hold actions” on page 734.
4. As **su**, run the Policy Renewal Start batch process as described in “Renew policies through batch process” on page 733.
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.

6. In the second commercial property policy, select **Actions**→**Policy Hold**→**Do not issue when hold released**.
7. Complete the activity.
8. As **su**, delete the policy hold:
 - a. Select **Administration**→**Business Settings**→**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. 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 Holds screen

Use the **Policy Holds** screen to administer policy holds.

Note: You must have sufficient permission to view, and work with the **Policy Holds** screen. For more information, see “Policy hold permissions” on page 739.

Create or edit a policy hold

Procedure

1. Navigate to **Administration**→**Business Settings**→**Policy Holds**.
 2. Choose to 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.

Policy 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 Regulatory Hold or Underwriting Hold . Your selection changes the availability of the UW Issue Type and Long Description 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 Risk Analysis 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 Hold Type selection determines the drop-down list choices. If the Hold Type is Underwriting Hold , then the only choice is UW Policy Hold . If the Hold Type is Regulatory Hold , then the only choice is Regulatory Policy Hold . The UW issue types are defined in the UWIssueType system table.
Long Description	Required. Provide a description for the policy hold. This field appears under the Description on the Risk Approval Details 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 Hold Type selection affects the drop-down list choices. If the Hold Type is Underwriting Hold , then the choices are Effective Date and Written Date . If the Hold Type is Regulatory Hold , then the only choice is Reference Date .

Field	Description
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 730
- “Determining the reference date for availability” on page 449

Policy Hold Regions tab

On the **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
Do not issue when hold released	Do not issue the policy when PolicyCenter releases the underwriting hold.
	When the Policy Hold Job Evaluation 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.
Allow issuance when hold released	This is the default setting. Issue the policy when the underwriting hold is released. When the Policy Hold Job Evaluation 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.

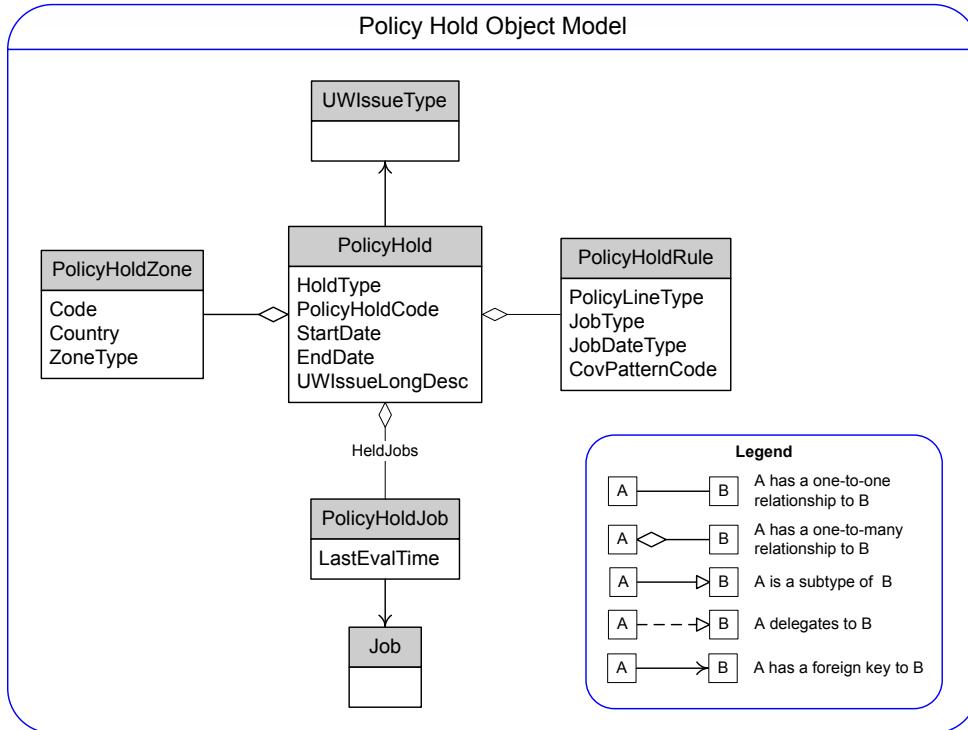
Note: If a renewal has been escalated to manual processing for another reason, then PolicyCenter does not return the renewal policy transaction to automated processing. This behavior applies even if the policy is set to **Allow issuance when hold released**. For example, assume the producer modified 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

- “Work with policy hold actions” on page 734
- “Policy Hold batch process” on page 740

Policy hold object model

The following illustration shows some of the objects related to policy hold.



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 739
- *Configuration Guide*

Configuring policy holds

Policy holds are a type of underwriting issue. This topic describes how to configure policy holds.

See also

- “Underwriting issues” on page 671
- *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 Policy Holds screen.
Create policy hold	polholdcreate	Permission to create a policy hold.

In the sample data, the **Underwriting Supervisor** role has all the policy hold permissions. The **Underwriter** role has the **View policy holds** permission. For information about loading the sample data, see 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 721

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** and selecting **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: **UWHold** and **RegulatoryHold**. 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 issues” on page 671
- 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.lib.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→Do not 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

- *System Administration Guide*

Importing and exporting policy holds

Administrators can import and export policies under a policy hold.

See also

- *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.

Note: ClaimCenter enables you to specify holidays by zone, such as state and zip code, for use when assigning activities by location. PolicyCenter does not provide support for zones.

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, on the PolicyCenter **Administration** tab, navigate to the **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.

The **Applies to All Zones** field has no effect in PolicyCenter. The application always behaves as if **Applies to All Zones** is **Yes**.

Add a holiday

Before you begin

To add a holiday, the PolicyCenter user must be logged in with administrator privileges.

Procedure

1. From the **Administration** tab, select **Business Settings**→**Holidays** to open the **Holidays** screen with its list of holidays.
2. Click **Add Holiday** to create a new holiday.
3. Enter the holiday name, date, and type.
- 4.
5. Click **Update**.

Edit a holiday

Procedure

1. From the **Administration** tab, select **Business Settings**→**Holidays** to view the **Holidays** screen and the list of holidays.
2. Select the holiday to edit by clicking its link in the **Holiday** column. The holiday's field values are shown.
3. Click **Edit**.
4. Edit the field values.
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.
5. Click **Update**.

Next steps

You might need to change the **Date** of some holidays annually.

Delete a holiday

Procedure

1. From the **Administration** tab, select **Business Settings**→**Holidays** to view the **Holidays** screen and the list of holidays.
2. Select the holiday to delete.
3. Click **Delete**.

Create a new zone or type

Procedure

1. In Guidewire Studio, navigate to the typelist that you want to modify.
 2. Edit the typelist fields.
 - **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.

Because PolicyCenter does not support zones, 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 744

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 an insurer begin on Monday and end on Saturday. For this insurer, 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.

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

- *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, an insurer promises to respond to all inquiries and claims within two hours after receiving an inquiry. You call the insurer on Friday at 4:30 p.m., and Monday is a holiday. The insurer 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.

While certain methods might appear to be similar, they can have different results.

- 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`

To determine which roles have this permission, refer to the Security Dictionary.

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.

Note: See 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.
Exclusion forms	Forms that limit coverages. For example, a Mold and Fungus Exclusion Form can limit coverage on a home-owner'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 insurer 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 insurers 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 insurer 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

- “Document management” on page 677
- *Product Model 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** – Infer forms related to quote and bind
- **As it binds the policy** – Infer forms related to bind

Note: However, in submission policy transactions, forms specified as inferred at bind are inferred at issuance instead. In submissions, forms are removed from the policy at bind. When you issue a policy by selecting **Issue Policy**, form inference determines which forms to add. If you select **Issue Policy** without first selecting **Bind Only**, the forms are first removed, then form inference determines which forms to add.

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 435 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**→**Business Settings**→**Policy Form Patterns** screen of PolicyCenter.

Search for form pattern

About this task

You can search for form patterns by form number or product.

Procedure

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.
- Select a number from the **Duplicate** drop-down list.
- Edit only the settings that are different from the original.

PolicyCenter enables the **Duplicate** drop-down if you selected exactly one form pattern. 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.

- Select the check box of one or more form patterns to delete.
- 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 this button. The code for this permission is `formpatdelete`.

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.

Add form pattern

Procedure

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 748 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.

Specify the removal or replacement form pattern for a policy change

Procedure

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?**

Specify the associated form for a policy change

Procedure

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 751

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 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.

This contains topics that describe tabs on the **Form Pattern** screen.

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.
IMPORTANT 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?	<p>PolicyCenter uses this field to infer forms either at quote or at bind.</p> <p>If you answer Yes, then the form is inferred at quote and bind. However, in submission policy transactions, the form is inferred at quote and issuance. In submissions, forms are removed at bind.</p> <p>If you answer No, the form is inferred only at quote.</p> <p>The Quote 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.</p> <p>For bind-time forms, the Bind 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.</p> <p>This field sets the <code>FormInferenceTime</code> typelist in <code>FormPattern.InferenceTime</code> to either quote or bind.</p>
Assign an endorsement number to this form?	<p>Answer Yes if this form must have an endorsement number assigned to it at inference time.</p> <p>The endorsement number indicates the order in which the form is added to the policy. The insurer decides whether the form requires an endorsement number.</p> <p>Some insurers 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.</p>
Priority	<p>An integer value that PolicyCenter uses to determine:</p> <ul style="list-style-type: none"> • The order for assigning an endorsement number • The processing order for forms during form inference <p>PolicyCenter processes and numbers the lower numbers first. PolicyCenter interprets a null value for this field as <code>MAX_INT</code>.</p>
Integration Fields→Reference Code	<p>Provided as a field for your customization. The base configuration of PolicyCenter does not use this field.</p> <p>You can use this field to specify a code for other systems in your enterprise to identify this form.</p>

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
- 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.

An audit policy transaction generates information that is non-contractual. Therefore audit policy transactions are not appropriate for forms, and audit is not a selection on the **Transaction Types** tab. Use documents for generating information in an audit policy transaction. See “Document management” on page 677.

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
Group Code	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 Group Code is specified, PolicyCenter uses the Form Number as the Group Code of the form.
This form is the jurisdiction-specific version of:	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.
Policy Types / Coverage Parts	Specify policy types or coverage parts. If no values are specified, matches any policy type or coverage part.
Form inference conditions	Values are: <ul style="list-style-type: none">• <none selected>• An associated form is invalidated – 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.• An associated form is invalidated or updated – In a policy change, add this form if an associated form:<ul style="list-style-type: none">◦ Is no longer valid.◦ Was replaced by an updated copy of the same form. For example, if the schedule of items printed on the form was updated.• No additional criteria – Always add this form as long as the product, policy transaction, and jurisdiction match.• No additional criteria - added every job – Always add this form as long as the product, jurisdiction, and policy types / coverage parts match.• Selected Additional Insured type is used – Add this form if at least one additional insured of the selected types exists on the policy. You can select one or more additional insured types. If no values are specified, then any selected value will infer this form pattern.• Selected Additional Interest type is used – Add this form if at least one selected additional interest of the selected types exists on the policy. You can select one or more additional interest types. If no values are specified, then any selected value will infer this form pattern.• Selected coverage, condition, or exclusion is not used – Add this form if the selected coverage, condition or exclusion does not exist on the policy.• Selected coverage, condition, or exclusion is used – 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.

You can also specify that a policy change will update the form if there are changes to any of the selected:

- Coverage terms
- Fields in covered objects

Field	Description
	<ul style="list-style-type: none"> • Selected covterm value is used – 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>You can also 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"> ◦ Fields in covered objects <ul style="list-style-type: none"> • Selected typelist value is chosen – 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: <ul style="list-style-type: none"> ◦ Any coverable on the selected policy line. ◦ Any typelist field on that coverable. ◦ Any value on that typelist. You can choose only one value. <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.GenericFormInference` interface.

You may have a form that requires inference logic beyond what you can specify in the **Inference** tab. In this case, you can either configure a new generic forms inference class or 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 757
- “Policy Change tab for form patterns” on page 751

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→Business Settings→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.

Adding a custom inference class for form patterns

This topic describes how 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`.

Follow these steps to add a form pattern that uses a custom inference class.

Define inference class

Procedure

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 “Configuring custom form inference” on page 756 for details on the methods you must create.

Next steps

“Add form to the custom form inference table in Product Designer” on page 755

Add form to the custom form inference table in Product Designer

Before you begin

“Define inference class” on page 754

Procedure

1. In Product Designer, open `custom_form_inference.xml` in **System Tables**.
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`.

Next steps

“Define the form pattern in PolicyCenter” on page 755

Define the form pattern in PolicyCenter

Before you begin

Complete the step “Add form to the custom form inference table in Product Designer” on page 755 before you perform this step.

About this task

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 the *Installation Guide*.

Procedure

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	<code>WC_00_06_03_KS</code>
Number	<code>WC 00 06 03 KS</code>

Field	Value
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 "Define the form pattern in PolicyCenter" on page 755 to see this value in the drop-down list.

12. Add any addition information about the form. Click **Update**.

Result

You are done adding a form pattern that uses a custom inference class.

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.xml` system table in Product Designer. 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 edit rows in the `custom_form_inference` table.

For each row in the table, you must enter:

Field	Description
FormCode	The code for the <code>FormPattern</code> . 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.

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 the *Integration Guide* for more information on forms inference.

Configuring form inference by coverage part and policy type

Through configuration, you can add form inference by coverage part and policy type to lines business. For an example, see the homeowners line. To add this type of form inference in a line of business:

- Add coverage part and policy types to the line by extending `CoveragePartType.tti` to include the coverage part and the policy types. For an example, see `CoveragePartType.HOP.tix`.
- For each coverable, implement the `getAssociatedCoveragePartTypes` getter in the `CoverableAdapter`.

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`.

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.

Administration utilities

This topic describes tasks that you access from the **Administration** tab's **Utilities** menu in PolicyCenter.

Importing and exporting data

While you enter much of the PolicyCenter administrative data directly into the application, there are times when it is necessary or more convenient to transfer this data in bulk. The **Import Data** and **Export Data** menu items of the **Administration**→**Utilities** tab provides a convenient way of moving data. You can use this feature to either import or export data into or from PolicyCenter in bulk.

See also

- *System Administration Guide*

Administering script parameters

A script parameter is an application-wide global parameter that has a value that tends to change over time.

Script parameters are defined in Guidewire Studio. Script parameters and their values are stored in the `ScriptParameters.xml` file.

After being defined, a script parameter can be modified in either Studio or through the PolicyCenter user interface.

- In the Guidewire Studio **Project** window, navigate to **configuration**→**config**→**resources** and double-click `ScriptParameters.xml` to open the file. Parameters can be created, deleted, and modified. Gosu code can reference a system parameter as a global variable.
- A PolicyCenter user must have system administration permission to modify a system parameter value in the user interface. In the **Administration** tab, navigate to **Utilities**→**Script Parameters** to view the list of parameters.

See also

- *Configuration Guide*

Administering data changes

In the **Administration** tab, the **DataChange** menu link enables you to push data changes to the production server. Use this feature sparingly and only to update mission-critical data on running production systems.

See also

- *System Administration Guide*

Importing and exporting policy data spreadsheets

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.

See also

- “Policy data spreadsheet import/export” on page 437
- *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.

Define export formats

About this task

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.

A format can only remove columns from the spreadsheet.

Procedure

1. Click the **Administration** tab and select **Utilities**→**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 Commercial Property Locations or Commercial Property Buildings . 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 Remove . To include columns, select them in the list on the left and click Add .

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.

See also

- *Configuration Guide*

External system integration

Billing system integration

The default configuration of PolicyCenter includes a completely functional integration with Guidewire BillingCenter. Alternatively, you can integrate PolicyCenter with another billing system of your choice. This topic describes how PolicyCenter integrates with any billing system generally and provides specific information about how PolicyCenter integrates with BillingCenter.

See also

- *Installation Guide*
- *Integration Guide*

Billing system integration overview

The integration between PolicyCenter and a billing system allows them to share information about accounts, policies, producer organizations and producer codes. The billing system is BillingCenter if you enable the integration. PolicyCenter and the billing system exchange information by using Guidewire plugins and standards-based web services. The Billing System plugin sends information from PolicyCenter to the web services published by the billing system. Conversely, a variety of web services that PolicyCenter publishes receive information from the billing system.

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 default integration with BillingCenter, either PolicyCenter or BillingCenter is a system of record for some shared information.

The default 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, an insurer 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.

Organizations and producer codes in PolicyCenter and billing system

When integrated with a billing system, PolicyCenter propagates producer codes to the billing system.

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 insurer 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 insurer organization. If not, then PolicyCenter sends a message asking BillingCenter to create the producer for the internal insurer organization.

If you want to share organizations and producer codes between PolicyCenter and the billing system, you must create them in PolicyCenter.

PolicyCenter gets commission plans from the billing system when the user creates or edits commission plans for a producer code. The call to the billing system is asynchronous. In the base implementation, if the billing system cannot be reached, PolicyCenter displays the commission plans that it currently has.

PolicyCenter gets agency bill plans from the billing system when the user creates or edits agency bill plans for an organization. The call to the billing system is asynchronous. In the base implementation, if the billing system cannot be reached, PolicyCenter displays the agency bill plans that it currently has.

See also

- *Integration Guide*

Producer organizations in PolicyCenter and BillingCenter

Producer organizations represent agencies or brokers. A producer organization produces policies through individual producers, such as agents.

In PolicyCenter, the producer organization is referred to as an organization. The user interface has an **Organization** screen, and the data model has an **Organization** entity.

In BillingCenter, the producer organization is referred to as a producer. The user interface has a **Producer** screen, and the data model has a **Producer** entity.

See also

- “Community model overview” on page 699

Multicurrency and producer organizations

You set the currency for producer organizations through agency bill plans. In a multicurrency system, you can select one or more agency bill plans on the **Organization→Agency Bill Plans** tab. Each agency bill plan specifies one or more currencies. For each producer organization, you can associate only one plan per currency. Therefore, if you select USD on Plan A, you cannot select USD on Plan B.

See also

- *Configuration Guide*

Multicurrency and producer codes

In PolicyCenter, producer codes are associated with commission plans. Each commission plan can offer one or more currencies. PolicyCenter retrieves the commission plans from the billing system. In a multicurrency system, each producer code can have one or more commission plans. For each producer code, you can select which currencies to use from each commission plan.

See also

- *Configuration Guide*

Accounts in PolicyCenter and billing system

When integrated with a billing system, PolicyCenter propagates accounts to the billing system.

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 commercial 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 to specify 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.

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

- “Related accounts” on page 305
- *Installation Guide*
- *Integration Guide*

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 the following information:

- **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.

Service tier for an account

In the base configuration, you set the service tier in PolicyCenter. If you are integrated with BillingCenter 8.0 (or later), PolicyCenter propagates the service tier on an account to the service tier on the BillingCenter account. Every time a submission is bound and issued in PolicyCenter, PolicyCenter sends the service tier on the account to the BillingCenter account.

See also

- “Service tier in accounts” on page 307
- *Integration Guide*

Multicurrency and accounts

A PolicyCenter account with policies and assets in single currency usually has a single account in the billing system. In contrast, a PolicyCenter account with policies and assets in a mix of currencies may correspond to a single multicurrency account in the billing system. Alternately, the multicurrency PolicyCenter account may have a set of affiliated accounts in the billing system, one for each currency in the PolicyCenter account.

Policies in PolicyCenter and billing system

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 commercial 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

- “Accounts in PolicyCenter and billing system” on page 767

Sending charges and other information to billing system

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.

Payment screen in PolicyCenter

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 to view data in BillingCenter.

Contacts in PolicyCenter and billing system

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.

Billing system and 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.

The following series of actions and messaging occur between the applications when you quote a policy for the following jobs: submission, renewal, and rewrite.

PolicyCenter action	Messaging	Billing system action
<ul style="list-style-type: none"> User quotes. User advances to Payment screen. 	<ul style="list-style-type: none"> PolicyCenter sends message to get billing options, installment plans, and invoicing plans. 	<ul style="list-style-type: none"> Check to see if the producer code of record allows agency bill for this producer. Look up installment and invoicing plans.
	<ul style="list-style-type: none"> Billing system returns billing options, available installment plans, and invoicing plans. 	
<ul style="list-style-type: none"> User selects billing method. User selects installment plan. (Optional) User clicks to preview payments that billing system returned. 	<ul style="list-style-type: none"> PolicyCenter sends message with the selected payment plan. 	<ul style="list-style-type: none"> Calculate payment schedule.
		<ul style="list-style-type: none"> Billing system returns payment schedule which can be viewed in PolicyCenter.
<ul style="list-style-type: none"> User selects invoicing plan. (Optional) User enters one or more up-front payments. User binds policy. 	<ul style="list-style-type: none"> If the billing system does not know about the account, PolicyCenter sends account information. If the user adds an invoicing plan, PolicyCenter sends new invoicing information. 	<ul style="list-style-type: none"> If account is unknown, create account. If new invoicing plan, add to invoicing plans.
	<ul style="list-style-type: none"> 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. 	
		<ul style="list-style-type: none"> Create policy or policy period on the account. Set <code>PolicyPeriod.BoundDate</code> to the policy's model date (the date when the policy was bound) as reported by <code>PolicyCenter.SetPolicyPeriod.TermConfirmed</code> to <code>false</code>. The policy term/period is

PolicyCenter action	Messaging	Billing system action
		bound, but not confirmed until payment is received and any other conditions for official binding are met.
		<ul style="list-style-type: none"> Process and apply the new charges. Process payment when received. Call the <code>IPolicyPeriod</code> method <code>hasReceivedSufficientPaymentToConfirmPolicyPeriod</code> to determine whether payment is sufficient. Payment may be sufficient if it is equal to or greater than the amount due on the renewal's first invoice. The call returns true if the payment is sufficient.
	<ul style="list-style-type: none"> If <code>PolicyPeriod.ConfirmationNotificationState == NotifyUponSufficientPayment</code>, BillingCenter notifies PolicyCenter that sufficient payment has been received by calling the PolicyCenter <code>PolicyRenewalAPI.confirmTerm</code> method. 	
<ul style="list-style-type: none"> Set <code>Policy.PolicyTerm</code> flag to true. 	<ul style="list-style-type: none"> PolicyCenter sends BillingCenter a message that the policy term/period is officially confirmed. 	<ul style="list-style-type: none"> Record the confirmation by setting <code>PolicyPeriod.TermConfirmed</code> to true.

See also

- “Working with the Payment screen” on page 775
- Integration Guide*

Billing system and policy transactions that create midterm changes

The issuance, policy change, cancellation, and reinstatement policy transactions (jobs) create midterm changes.

The following series of actions and messaging occur for the applications when you quote a policy for job types issuance (not submission), policy change, cancellation, and reinstatement.

PolicyCenter	Messaging	Billing system
<ul style="list-style-type: none"> User quotes in PolicyCenter 		
<ul style="list-style-type: none"> User advances to Payment screen. This step applies to most jobs, but not Cancellation. A payment through the payment gateway can also be made for changes and other transactions. 		
<ul style="list-style-type: none"> User binds policy. 		

PolicyCenter	Messaging	Billing system
	<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • Billing system actually processes and applies new charges.

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

- *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

- *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 insurer 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 insurer 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 insurer 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 773

Reinstatements in the billing system integration

When the insurer receives payment after canceling a policy, the insurer may choose to reinstate the policy. The insurer’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 insurers, 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

- *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.

See also

- For more information on conversion on renewal process, see 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 items just in case the billing system changed them during the current period. PolicyCenter displays these fields on the **Payment** screen 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 “Cancellations in the billing system integration” on page 772 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

- *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

- *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

- *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** screen.

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 contains topics that describe some of the changes in the user interface when you enable an integration with a billing system such as BillingCenter.

See also

- *Installation Guide*

Working with the Payment screen

The **Payment** screen appears in a submission, policy change, renewal, or rewrite policy transaction.

The **Payment** screen displays payment information for the policy. You view the **Payment** screen after quoting but before binding the policy. When integrated with a billing system, PolicyCenter retrieves billing methods, payment methods, and associated installment or reporting plans from the billing system. After you select a payment plan, 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. When the policy is bound or issued, PolicyCenter passes the job number and billing information to the billing system. When the payment (from the payment gateway) reaches the billing system, the user can track the payment by job number. In PolicyCenter, the payment appears in the **Collected** table. 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

- *Installation Guide*
- *Integration Guide*

Premium Summary on Payment screen

At the top, the **Payment** screen displays summary information in the **Premium Summary**. The summary shows values associated with premium, including:

- **Total Premium**
- **Taxes and Surcharges**
- **Fees**
- **Total Cost** – The sum of the previous fields

Payments

The **Payments** summary displays information from **Billing** and **Payment Schedule**.

Invoicing

The **Invoicing** summary displays information from **Invoicing Overrides** if any are specified. Otherwise **Invoicing** displays default values.

Override Change in Cost Billing on Payment screen

Override Change in Cost Billing appears in policy change, issuance, and reinstatement policy transactions. These fields do not appear for workers' compensation.

Note: If you are integrated with BillingCenter version 9.0, this field does not appear. PolicyCenter displays **Billing and Issuance Options** instead.

The **Special Handling** field enables you to override how to bill the changes in cost. The options are:

Bill Immediately

Charges are billed immediately. If an invoice for the current date already exists, the charges are added to the existing invoice, otherwise a new invoice is created.

Bill on Next Invoice

Charges are added to the next invoice.

Hold for Final Audit

Appears for policies that have the schedule final audit set to **Yes**. The charges are held until the final audit. This is typically used when the premium change is so small that the insurer prefers to wait for the final audit to invoice the customer for the changed premium. Setting this value creates a charge in BillingCenter, but BillingCenter puts the charges on hold.

This information can be sent to the billing system. When integrated with BillingCenter, the information is sent to BillingCenter.

Billing and Issuance Options

If you are integrated with BillingCenter version 9.0, **Billing and Issuance Options** appears in policy change, issuance, and reinstatement policy transactions if there is a change in costs.

Select **Override billing allocation** to change how the costs are billed:

- **Bill immediately** – Specify the percentage of the charges to bill immediately.
- **Allocation of remainder** – Specify how to bill the remaining charges.

Billing on Payment screen

You can choose billing information in a submission policy transaction. The **Billing** heading on the **Payment** screen contains the following fields:

- **Billing Level** – This value is not editable. Values are:
 - **Account** – Group policies together into a single account bill
 - **Policy (Separate Funds by Policy)** – Bill each policy individually with cash separation
 - **Policy (Separate Funds by Account)** – Bill each policy individually without cash separationThe billing level is retrieved from the account in the billing system. In the integration with BillingCenter, the value comes from the account in BillingCenter. If it is the first submission on a new account, the value comes from the **BillingSystemPlugin** in PolicyCenter.
- **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. 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.

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 a **Payment Plan**. Billing is negotiated between the agent and the insurer. Therefore, the screen hides the **Alt Billing Account** selection.

If you select an alternate billing account, the policy gets the billing level from the alternate account.

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**.

Payment Schedule on Payment screen

Under the **Payment Schedule** heading on the **Payment** screen, you can select an installment or reporting plan and override invoicing. The installment or reporting plans are retrieved from the billing system.

The **Payment Schedule** lists payment plans which specify numbers of payments and a down payment percentage. Each **Payment Plan** is retrieved from a billing system. If you are integrated with the **StandAloneBillingSystemPlugin**, the plugin returns a list of sample payment plans. If you are integrated with BillingCenter, the plugin returns a list of valid payment plans for the selected list bill payer. The integration retrieves payment plans from BillingCenter.

Note: If you change the language while viewing the **Payment** screen, the language is not updated for **Payment Plan** items. To refresh the language, navigate away from and back to this screen.

Payment Schedule popup

For the selected **Payment Plan**, click the image in the **Schedule** column to open a **Payment Schedule** popup with payment amounts and due dates generated by the billing system. The **Charges for** column displays charges for the current policy transaction. Click **Expand All** to view details.

Show all charges on invoices enables you to see a comprehensive view of the customer's invoices. This button appears if the current policy will share invoices with other policies after binding. Select this to view additional columns. The **Other Policies on Invoice** column includes charges for other policies that will be on the same invoice. The **Total on Invoice** amounts is the total amount for all policies on the invoice. For policy changes, the **Current Charges for this Policy** column displays charges for the policy prior to this policy change. **Charges for this Change** displays charges for this policy change.

If the billing level is account, the **Other Policies on Invoice** column includes charges for other policies that will be on the same invoice. The **Total on Invoice** amounts is the total amount for all policies on the invoice. For policy changes, the **Current Charges for this Policy** column displays charges for the policy prior to this policy change. **Charges for this Change** displays charges for this policy change.

If the billing method is agency bill, then for each payment, the **Due Date** is calculated using the selected payment plan. The due date is not calculated using the agency bill plan.

Invoicing Overrides

You can override invoicing if the billing level is on the policy. If the billing level is on the account, this field does not appear. The **Invoicing Overrides** label is appended with the name of selected payment plan. The fields vary depending upon the payment plan. Select **Invoicing Overrides** to display invoicing fields:

Fix Invoices by

Bill Date specifies that invoices are sent on the specified **Day of Month**. **Due Date** specifies that the invoice is due on the specified **Day of Month**. In the integration with BillingCenter, the due date or invoice send date are calculated using the lead time on the billing plan in BillingCenter.

Day of Month

If **Monthly** or **Twice per Month** is selected, the day of the month to send invoices or the due date.

Second Day of Month

If **Twice per Month** is selected, the second day of the month to send invoices or the due date.

First Payment Date

If **Every Other Week** is selected, the day of the month to send the first payment.

Description

Optional field to describe the override.

Pay Using

Select or add a payment method.

Click the **Add** button to set up a new payment method in a payment 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 payment system, clicking the **Add** button could take the user to a similar screen presented by the payment system. After entering the data, the user is returned to PolicyCenter. For information on integrating with a payment system, the *Integration Guide*.

These invoicing overrides appear in the **Invoicing** summary at the top of the **Payment** screen.

Up-front Payments on Payment screen

In **Up-front Payments** on the **Payment** screen, you can enter the amount of money collected by the 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.

The **Electronic** payment enables you to use a payment gateway to retrieve electronic payment options. In the base configuration, the payment gateway is a standalone plugin.

- Select a payment instrument to be redirected to the payment gateway. The list of payment instruments is obtained from the billing system.
- The amount suggested is the down payment minus the amount already paid. This is the same for Held by Agent, Check and Cash.

See also

- *Integration Guide*

View policy period in BillingCenter

About this task

If the BillingCenter integration is enabled, you can view the policy period in BillingCenter after binding the policy.

Procedure

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 **Account→Billing** screen 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 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 enables 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. If you are in a multicurrency system, the link jumps to the primary affiliated account in BillingCenter.

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 765

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.

Multicurrency integration between BillingCenter and PolicyCenter

BillingCenter and PolicyCenter support integration of multicurrency accounts and producers. PolicyCenter manages the relationship aspects of accounts and producers. BillingCenter manages the accounts receivable, commissions payable, and cash application aspects of accounts and producers. Because the two applications manage different aspects of accounts and producers, each application represents them differently for multicurrency purposes.

PolicyCenter manages the relationship between the insurer and an insured through policies. An account in PolicyCenter represents this relationship. PolicyCenter permits an account to have policies that cover assets valued in different currencies. For example, a commercial property policy might cover properties in multiple countries, with each property on the policy valued in the currency of its location.

BillingCenter manages billing activities for policies, and it pays commissions earned by producers of policies. To maintain accounting integrity, BillingCenter does not permit an account to have transactions in multiple currencies. Instead, BillingCenter manages each account to be certain that transactions can occur in only one currency. Similarly, BillingCenter tracks commission earned by each producer in only one currency.

Therefore, while PolicyCenter has a single account or producer regardless of currencies, BillingCenter must have multiple affiliated accounts and producers, one for each currency.

Limitations of multicurrency billing and policy integration

Multicurrency integration between BillingCenter and PolicyCenter has the following limitations in the base configuration.

- **Delinquency Processing** – In the base configuration of BillingCenter, each affiliated account has its own delinquency process. If one affiliated account becomes delinquent, the other affiliated BillingCenter accounts do not become delinquent automatically.
- **List Bill Accounts** – In the base configurations of BillingCenter and PolicyCenter, list bill accounts do not support multicurrency integration. List bill accounts are payers for policies in a single currency only.

You can overcome the preceding limitations with configuration.

Multicurrency accounts in PolicyCenter and BillingCenter

A PolicyCenter account with policies and assets in single currency has a single BillingCenter account. In contrast, a PolicyCenter account with policies and assets in a mix of currencies has a set of affiliated BillingCenter accounts, one for each currency in the PolicyCenter account. The currencies in a PolicyCenter account are the currencies in all policy periods of all policies associated with the account. PolicyCenter communicates solely with the primary affiliated account. In a multicurrency system, the integration with BillingCenter sends communications to the appropriate affiliated account.

Within a set of BillingCenter accounts affiliated with a PolicyCenter account, one account is the primary affiliated account. A *primary affiliated account* retains the public ID of its correspondent multicurrency account in PolicyCenter. PolicyCenter retains the public ID only of the primary affiliated account and remains unaware of any secondary affiliated accounts created by BillingCenter.

Whenever PolicyCenter sends integration messages to BillingCenter that involve multicurrency accounts, PolicyCenter passes a currency parameter and the public ID of the account in BillingCenter that PolicyCenter first created. BillingCenter uses the currency that PolicyCenter passes to determine whether to:

- Split the account into a primary affiliated account and a secondary affiliated account for the new currency.
- Create an additional affiliated account for the new currency.
- Locate the affiliated account for the currency.

Regardless of the actions that BillingCenter takes, PolicyCenter is aware of only one account in BillingCenter.

See also

- “Account overview” on page 305

Using commission plans to select currencies for producer codes

In BillingCenter and PolicyCenter, producer codes identify individual producers.

Individual producers earn commissions on the policies they produce. In some cases, producers share the same producer code. PolicyCenter tracks which producer codes produced which policies. BillingCenter tracks commissions earned by each producer code and periodically initiates commission payments. Commissions generally are not paid directly to individual producers. Instead, aggregate commissions are paid to the producer organization, which in turn disburses commission payments to individual producers.

In PolicyCenter, you add commission plans to producer codes. In BillingCenter, you define the commission plans that you can add to producer codes in PolicyCenter.

In BillingCenter, a commission plan can offer multiple currencies. In PolicyCenter, you select the currencies you wish to use from each commission plan. For each producer code, you can associate only one plan per currency.

Therefore, if you select USD in commission plan A, you cannot select USD in commission plan B. If a commission plan is in use for that producer code, you cannot change the currency selections. After making changes,

PolicyCenter sends the new or updated producer organizations and producer codes to BillingCenter, along with the commission plans and selected currencies.

Single currency producer codes in PolicyCenter and BillingCenter

In a single currency system, a producer code in PolicyCenter has one commission plan. This commission plan has a single correspondent producer code in BillingCenter. All commissions are earned and paid in that currency. The producer code in BillingCenter retains the public ID of its correspondent producer code in PolicyCenter, and the producer code in PolicyCenter retains the public ID of its BillingCenter counterpart. In addition, the producer code in BillingCenter is associated with its BillingCenter producer in the same currency.

Multicurrency producer codes and producer organizations

A producer code in PolicyCenter with commission plans in a mix of currencies has a correspondent set of affiliated producer codes and producer organizations in BillingCenter. In BillingCenter, each affiliated producer organization aggregates commissions earned by its producer codes in one of the currencies of the organization's commission plans. Each affiliated producer code for a specific currency is associated with its affiliated producer organization of the same currency.

Implementation details

Within a set of BillingCenter affiliated producer codes associated with a PolicyCenter producer code, one producer code is the primary affiliated producer code. A *primary affiliated producer code* retains the public ID of its correspondent multicurrency producer code in PolicyCenter. PolicyCenter retains the public ID only of the primary affiliated producer code and remains unaware of any secondary affiliated producer codes created by BillingCenter.

Whenever PolicyCenter sends integration messages that involve multicurrency producers codes, PolicyCenter passes a currency parameter and the public ID of the producer code in BillingCenter that PolicyCenter originally created. BillingCenter uses the currency that PolicyCenter passes to determine whether to perform the following actions.

- Split the producer code into a primary affiliated producer code and a secondary affiliated producer code for the new currency.
- Create an additional affiliated producer code for the new currency.
- Locate the affiliated producer code for the currency.

Regardless of the actions that BillingCenter takes, PolicyCenter remains aware of one producer code only in BillingCenter.

Default billing and policy multicurrency integration

The default implementations of plugins and web services in BillingCenter and PolicyCenter handle the differences in multicurrency accounts and producers automatically. However, you must configure both application instances to enable their multicurrency integration successfully. For example, you must set up each instance with the currencies they share.

See also

- *Configuration Guide*

Claim system integration

The base configuration of PolicyCenter enables you to configure a completely functional integration with Guidewire ClaimCenter. You can also integrate PolicyCenter with the claim system of your choice.

See also

- *Installation Guide*
- *Integration Guide*

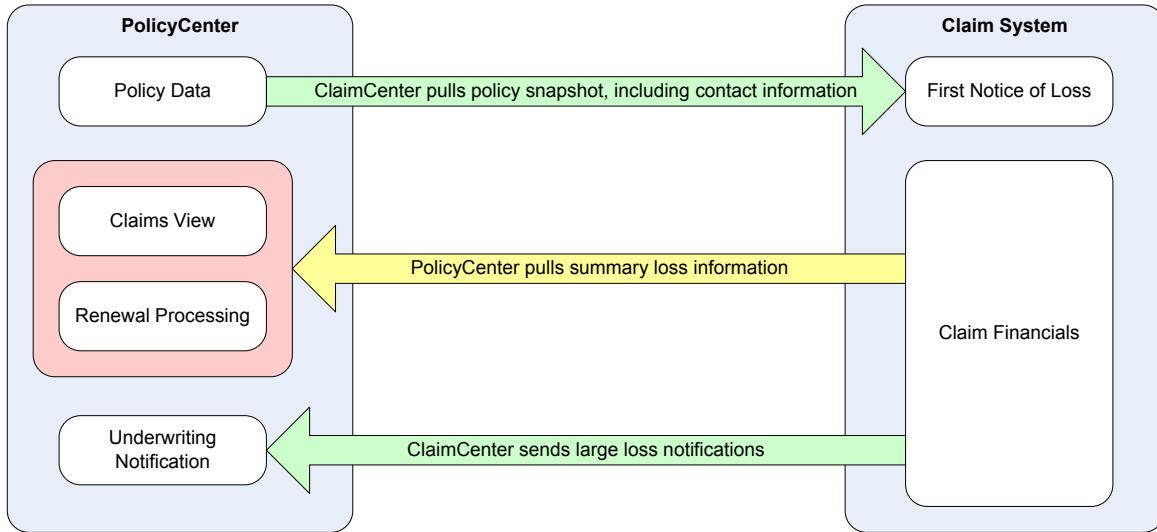
Claim system integration overview

You can configure PolicyCenter to retrieve claims from a claim system. The base configuration provides a completely functional plugin implementation class that supports 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.



The parts of the integration that originate in ClaimCenter are:

- ClaimCenter pulls policy snapshot.
In the base configuration, if you are integrated with ClaimCenter 8.0 (or later), the snapshot includes the service tier on the PolicyCenter account. The service tier is propagated to the service tier on the ClaimCenter policy.
- ClaimCenter sends large loss notification

See also

- “Service tier in accounts” on page 307
- *Installation Guide*
- *Integration Guide*

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:

- | | |
|--|---|
| <ul style="list-style-type: none"> • Claim number • Loss date • Loss cause • Status • Description | <ul style="list-style-type: none"> • Information on injuries • Value for remaining reserves • Value for total paid • Value for total recoveries • Value for net total incurred |
|--|---|

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:

- | | |
|--|--|
| <ul style="list-style-type: none">• Claim Type• Claim Number• Policy Period Start• Policy Period End• Loss Date• Loss Cause• Status | <ul style="list-style-type: none">• Description• Litigation• Injuries• Remaining Reserves• Paid to Date• Total Incurred• Recoveries |
|--|--|

Viewing claim on policy 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"> • Claim total incurred > \$10,000 • Ratio of claims total incurred to policy written premium > .5 (.5 = 50%) • Incidence of claims > .0005 (5 claims / \$10,000 of premium)
Underwriter 1 & Underwriter 2	<ul style="list-style-type: none"> • Claim total incurred > \$20,000 • Ratio of claims total incurred to policy written premium > .90 (.90 = 90%) • Incidence of claims > .0010 (10 claims / \$10,000 of premium) • Manual claim review needed • Unable to retrieve claim information
Underwriter Supervisor	<ul style="list-style-type: none"> • Claim total incurred: no limit • Ratio of claims total incurred to policy written premium: no limit • Incidence of claims: no limit • Manual claim review needed • Unable to retrieve claim information

See also

- “Underwriting issues” on page 671

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 from ClaimCenter

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 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.

See also

- *Integration Guide*
- *Guidewire 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 “New contact when integrated with contact management system” on page 792.

[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 and contact management system](#)” on page 790
- “[Pushing new and updated contacts to contact management system](#)” on page 791

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 **Contacts** 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 contact types are the **ABPerson** and **ABCCompany** subtypes of **ABContact**.

[See also](#)

- “[Adding a contact from the address book](#)” on page 363
- [Guidewire Contact Management Guide](#)

New and updated contacts and contact management system

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 to contact management system

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. In addition, changes to a policy contact might not be immediately synchronized with the account contact.

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

- For information on when PolicyCenter pushes a new contact to ContactManager, see “Adding a contact from the contact management system” on page 791
- For information on when changes to a policy contact are synchronized with the account contact, see “Revisioning contact information in policies” on page 355
- *Guidewire 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 363

New contact when integrated with contact management system

If a new customer calls to add a new driver to a personal auto policy, the agent creates a new contact for the customer in PolicyCenter.

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

- “Create a new contact” on page 358

Information required for saving contact 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 that is a person or a company.

ContactManager requires the following fields for a contact that is a person:

- **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

See also

- *Guidewire Contact Management Guide*

Information required for selecting contact from contact management system

In the base configuration, an external contact must have certain fields to be selectable in PolicyCenter. To **Select** a contact from an external contact management system, 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

- “Create a new contact” on page 358 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

- *Guidewire Contact Management Guide*

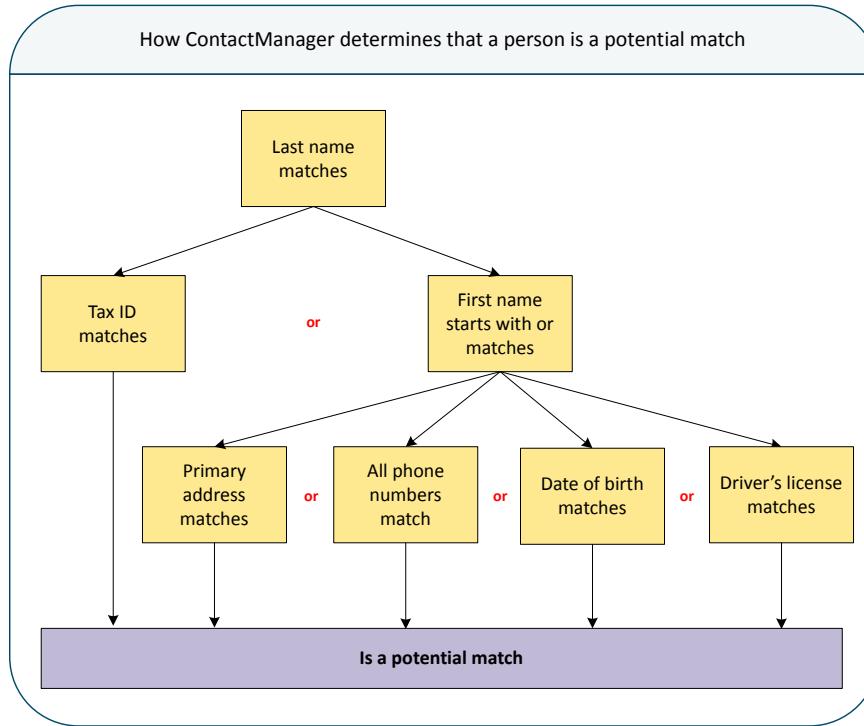
How ContactManager determines duplicate contacts for a person

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.

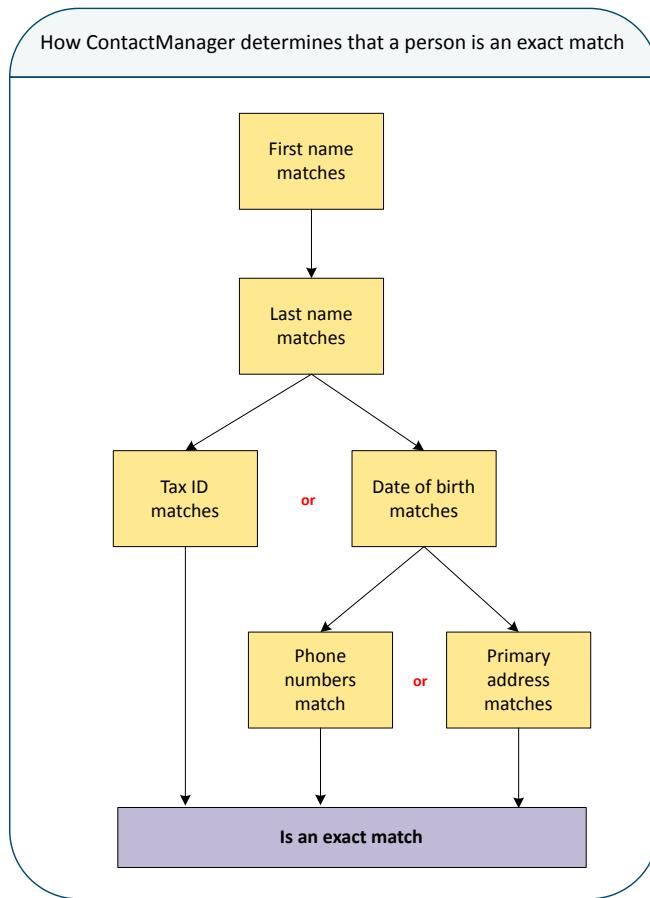
ContactManager first looks for a potential match. The following illustration shows the various paths through which ContactManager determines that a person is a potential match.



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.

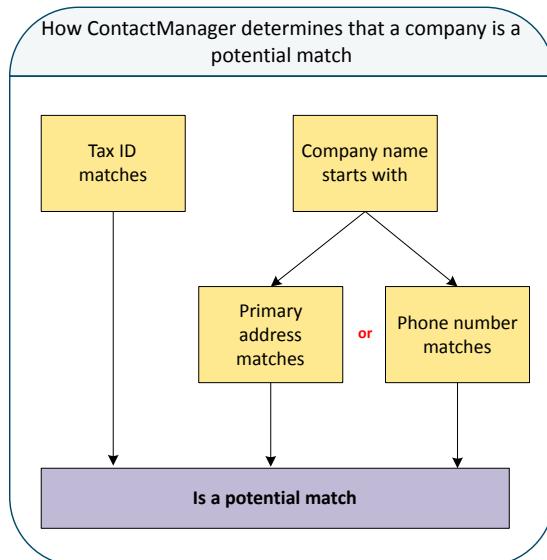


How ContactManager determines duplicate contacts for a company

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.

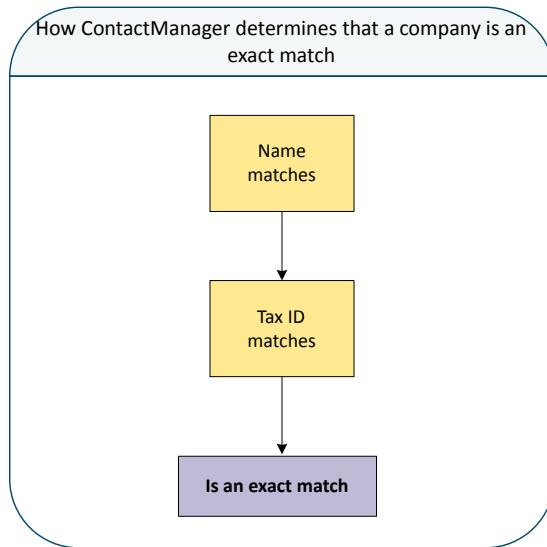
ContactManager finds a potential match if it finds a potential match through any path. The following illustration shows how ContactManager determines that a company is a potential match.



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`.

See also

- *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 policy transaction 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 deactivating 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.

See also

- *Integration Guide*
- *Guidewire Contact Management Guide*

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. See the *Integration Guide*.
- Search for contacts stored in a contact management system. See the *Guidewire Contact Management Guide*.
- Push new contacts from PolicyCenter to a contact management system. See the *Guidewire Contact Management Guide*.
- Push or pull updates to contacts from a contact management system to PolicyCenter. See the *Guidewire Contact Management Guide*.
- Push or pull updates to contacts from PolicyCenter to a contact management system. See the *Guidewire Contact Management Guide*.
- View all places that use a particular contact.
- List policies, accounts, and policy transactions associated with a contact.
- Merge contacts in PolicyCenter that the contact management system identifies as duplicates. See the *Guidewire Contact Management Guide*.

Contact management 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.
- **Policy transaction** – Given a policy transaction number, open that policy transaction directly from an external system.

See also

- *Integration Guide*

Integrating with multiple contact management systems

You can integrate PolicyCenter with multiple contact management systems. ContactManager can be one of these systems.

You can use different systems to store different types of contact information. For example, you might use one system for customer information such as named insured and drivers. You 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.

See also

- *Guidewire Contact Management Guide*