ORACLE
Insurance

Oracle Insurance

Insbridge Enterprise Rating

RateManager User Guide

Modules

Volume 1 Program Explorer

Release 5.6

November 2018

Copyright © 2005, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle Insurance Insbridge Enterprise Rating RateManager User Guide

Release 05.06.xx

Part # E96987-01

Library #E96993-01

November 2018

Primary Author: Mary Elizabeth Wiger

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.
Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

CONTENTS

PREFACE

Audience xxvii
Related Documents xxvii
Conventions xxvii
Navigating RateManager xxviii
Basic Safety Precautions xxviii
Inactivity/Timeout xxviii
System Requirements xxviii
Guide History xxviii

GETTING STARTED

Data Preparation and Program Structure 31
Restrictions 33
Naming Restrictions 37
Unicode 39
Logging Into the System 40
RateManager Tabs and Menus 41
NOTICE FOR UPGRADE USERS 46
Previous Releases 46
Release 5.0 and After 47
Program Structure Before and After 48

CHAPTER 1

RATEMANAGER CONSTRUCTS 49
Order of Operations 49
Locale 49
Date Masks 51
Number Formatting 52
Default Values 53
Parent Child Relationships 53
Versioning 54
Program Versioning 54
Element Revisions 56

CHAPTER 2

HOME TAB 57
Adding Snaps to the Home Tab 58
Changing Your Password 61

CHAPTER 3

PROGRAM EXPLORER 62

Program Explorer Hierarchy 62
Classification of Data 64
Creating Your Own Classification 64
Program Explorer Structure 68
Program Explorer Functionality 69
Programs 71
Programs Screen Details 71
Program Details 73
Program Details Screen Details 73
Program Creation 76

CHAPTER 4

SCENARIO SANDBOX 77
Scenario Update Required 79
Scenario Permissions 80
Scenario Creation Process 83
Creating a Scenario 84
Adding Programs to a Scenario 84
Removing a Program 87
Adding Tables to a Scenario 88
Removing a Single Table from a Scenario 89
Removing a Linked Variable Table from a Scenario 90
Editing a Scenario 92
Publishing a Scenario 94
Unlocking a Scenario 98
Deleting a Scenario 99
Scenario User 101
Scenario User – Program Details 102
Scenario User – Editing a Table 103

CHAPTER 5

DOMAIN TYPES AND DOMAINS 104
Domain Types 104
Other User Created Domain Type 105
Scenario Domain Type 105
New Other Domain Type 106
Editing an Other Domain Type 107
Deleting an Other Domain Type 108
Domains 110
New Domains 110
Editing Domains 111
Clearing Domains 113
Deleting Domains 115

CHAPTER 6

PROJECTS AND PRODUCTS 117
Projects 117
New Projects 118
Editing Projects/Selecting Products 119
Clearing Projects 122
Moving Projects 123
Deleting Projects 125
Products 127
Global Versioning Report 127
Category Visibility 128
Using Category Visibility 129

CHAPTER 7

FOLDERS 131
Manage Program Folders 132
Manage Program Folders Functionality 133
Manage Program Folders Screen Details 133
Creating a Folder 135
Renaming a Folder 136
Deleting a Folder 137
Editing a Folder 139
Moving a Folder 141
Creating a Program Version Folder 143
Renaming a Program Version Folder 144
Deleting a Program Version Folder 145

CHAPTER 8

PROGRAMS 146
For more on Versioning Criteria, please see Versioning Criteria 147
Versioning Criteria Functionality ..... **Error! Bookmark not defined.**
Versioning Criteria ..... **Error! Bookmark not defined.**
Creating a Program 153
Moving a Program 154
Editing a Program 155
Deleting a Program 156

CHAPTER 9

PROGRAM VERSIONS 157
Program Version Types 158
Creating a Program Version 158
Copying a Program Version 159
Deleting a Program Version 160
Editing a Program Version 162
Versioning Criteria 163
Versioning Criteria Functionality 164

Versioning Criteria 165
Program Versioning Data 168
Program Versioning Data Functionality 169
Program Versioning Data Screen Details 169
Managing Program Versions 171
Program Version Names 171
Entering a Program Version Name 172
Program Version Folders 173
Assigning a Program Version Folder 175
Unassigning a Program Version Folder 177
SoftRater Automatic Version Selection 178

CHAPTER 10

PROGRAM DETAILS 180
Program Details Functionality 181
Program Detail Folders 183
Program Detail Folders Screen Details 183
Folder Detail 183
Folder Detail Screen Details 183

CHAPTER 11

GLOBALS AND UNIVERSALS 197
Universal Elements 197
Working with Universal Variables 197
Universal Tab 198
Universal Tab Functions 199
Manage Project Relationships 200
Granting Access 202
Denying Access to Project Relationships 206
Global Elements 208
Working with Global Variables 208
Globals Tab 209
Globals Tab Functions 210

CHAPTER 12

CATEGORIES 211
When to Use a Category 211
Working with Categories 212
Categories Functionality 214
Categories Details 214
Creating a Category 215
Editing a Category 216
Deleting a Category 219

CHAPTER 13

FIELDS 220
Fields Functionality 221
Field Details 223
Creating a Field 223
Custom XML ID 225
Adding a Field from another Area 225
Editing a Field 227
Copying a Field 228
Deprecate a Field 229
Restoring a Field 230
Deleting a Field 231

CHAPTER 14

VALIDATIONS 232
Validations Functionality 233
Validations Details 234
Validations Setup 235
List of Values Tables 237
Using a List of Values 238
List of Values Rules 238
Creating a List of Values 239
Adding a List of Values to a Field 242
Adding a Field to a Variable 244
Editing Table Data with List of Values Fields 246
Changing the View 247
Working with List of Values Data in Test Case Editor 248
Editing a List of Values 249
Copying a List of Values 251
Setting the Active Revision of a List of Values 252
Deleting a List of Values 253

CHAPTER 15

RESULT VARIABLES 254
Result Variables Functionality 256
Result Variables Details 256
Creating a Result Variable 257
Adding a Result Variable from another Area 258
Editing a Result Variable 259
Copy a Result Variable 260
Deleting a Result Variable 261

CHAPTER 16

TABLE VARIABLES 262
Table Variable Options 262
Table Variables Functionality on Program Details Tab 264

Table Variable Tab 265
Table Variable Tab Functionality 266
Table Variable Tab Screen Details 266
Table Variables Linked Variables Listing 268
Table Variables Variable Information 269
Table Variables Available Variables 271
Table Variables Variable Selection Criteria 272
Variable Selection Criteria 274
Table Variables Masking 274
Table Variables Wildcard 275
Table Variables Interpolation 275
Creating a Table Variable 277
Linked Variables 279
Editing a Table Variable 280
Table Variable Copy Options 280
Copying a Table Variable 285
Deleting a Table Variable 286
Working with Table Data 287
Import Data 287
Data Preparation 288
Restricted Use of Special Characters 288
Working with Tab Delimited Files 290
Importing a Tab Delimited File in to a Table Variable 290
Working with MS Excel Files 295
MS Excel File Import Data File Options 297
Importing a MS Excel File in to a Table Variable 298
Export Data 309
Export Data and Wildcards 310
Export Data Tab-Delimited 311
Exported Tab Delimited Data and Unicode 312
Export Data to MS Excel Workbook 313
Export Data File 317
Overwriting Existing Files with Export to MS Excel 317
Editing an MS Excel File 318
Table Composition 318
Editing Table Data 321
Edit Data 322
Editing Data in a Cell 324
Editing Rows 325
Data Sources 338
Data Sources Grouping 339
Creating 340
Importing Data Sources Data 342
Editing Data Sources Data 343

Editing Data in a Data Source 346
Editing Rows of a Data Source 350
Deleting a Data Source 359
Working with Data Sources 360

CHAPTER 17

WORKING WITH STEPS 361
Entering a New Step 363
Editing Steps 364
Copy and Paste Steps 364
Viewing and Clearing the Clipboard 367
Setting the Next Step 369
Using Results of Step # Logic 369
Updating Terms 372
Comparison Operators 373
Setting Rounding 374
Decimal Precision 375
Where Clause 375
Conditions 377
OR Condition 378
AND Condition 378
Parentheses 379
Adding a Where Clause 382
Editing a Where Clause 382
Deleting a Where Clause 383
Adding Conditions 383
Editing Conditions 384
Deleting Conditions 385
Adding Parentheses 385
Editing Parentheses 386
Deleting Parentheses 386
Moving Steps 387
Deleting Term 388
Deleting Steps 390
Category Items 391
Get Category Item Step 391
Set Category Item Step 394
Get Ranked Category Item Step 396
Set Ranked Category Item Step 396
Count across Category Step 397
Data Type Functions 400
IsAlpha Step 400
IsDate Step 400
IsNumeric Step 401
Date Functions 401

Date Addition Step 402
Date Difference Step 402
IF Statements 403
Complex IF 405
Mask Step 409
Math Functions 409
Arithmetic Step 410
Adding a New Element 410
Adding Arithmetic Operators 411
Arithmetic Operators 411
Absolute Value Step 412
Power Step 412
Log Step 413
Log10 Step 413
Natural Log 414
Exponent Step 414
Rand Step 414
Fact Step 415
Square Root Step 415
Ceiling Step 416
Floor Step 416
Even Step 416
Odd Step 417
Trig Functions 417
Cos Step 418
Cosh Step 418
Acos Step 418
Acosh Step 419
Sin Step 419
Sinh Step 420
Asin Step 420
Asinh Step 420
Tan Step 421
Tanh Step 421
Atan Step 422
Atanh Step 422
Degrees Step 422
Radians Step 423
Statistical Functions 423
Log Gamma 423
Incomplete Gamma 424
Gamma 424
Ranking 425
Clear Ranking Step 425

Rank across Category Step 426
String Functions 426
Get Length 426
Set Message 427
String Addition 427
Sum across Category Step 428
Multiply Across Category 430
Minimum/Maximum Step 431
Set Underwriting to Fail Step 432
Set Principal Operator Variable 434
Re-calculate Vehicle Usage Step 434
Working with Data Queries 436

CHAPTER 18

CALCULATED VARIABLES 439
Calculated Variable Options 439
Calculated Variable Functionality Program Details Tab 441
Calculated Variable Tab 442
Calculated Variable Functionality 443
Calculated Variable Screen Details 443
Calculated Variable Information 444
Calculated Variable Details 447
Calculated Variable Available Variables, Functions and Constants 448
Calculated Variable Report 450
Creating a Calculated Variable 451
Editing a Calculated Variable 453
Calculated Variable Copy Options 454
Copying a Calculated Variable 457
Deleting a Calculated Variable 459
Resolving Dependencies 460

CHAPTER 19

ALGORITHMS 461
Universal, Global and Local Algorithms 461
Results from Algorithms 462
Algorithm Rules 462
Algorithm Options 463
Algorithm Functionality Program Details 464
Algorithm Tab 466
Algorithm Tab Functionality 467
Algorithm Screen Details 467
Algorithm Tab Algorithm Information 468
Algorithm Tab Algorithm Details 470
Algorithm Tab Available Variables, Functions and Constants 471
Algorithm Tab Algorithm Report 473

Creating an Algorithm 473
Using an Advanced Option 476
Create Category Instance Function 478
Creating a Category Instance 479
Editing a Create Category Instance 483
Deleting a Create Category Instance 483
Dynamic Category Instance Indicator 484
Creating a Category Mapping 485
Message Templates in Use 490
Editing an Algorithm 491
Algorithm Copy Options 492
Copying an Algorithm 494
Deleting an Algorithm 495

CHAPTER 20

DRIVER ASSIGNMENT 497
Global and Local Driver Assignments 497
Separating Driver and Vehicle Output (Auto Product Only) 498
Driver Assignment Options 500
Driver Assignment Functionality Program Details 501
Driver Assignment Tab 503
Driver Assignment Rules 503
Driver Assignment Tab Functionality 504
Driver Assignment Screen Details 505
Driver Assignment Tab Information 506
Vehicle Usage Options 507
Driver Assignment Tab Algorithm Details 508
Driver Assignment Tab Available Variables, Functions and Constants 510
Driver Assignment Tab Scenario or Algorithm Report 512
Scenario Functions 513
Flagging Functions 514
Available Flagging Functions 515
Flagging Algorithms 517
Ranking Functions 518
Ranking Algorithms 520
Assignment Functions 521
Miscellaneous Functions 525
Available Miscellaneous Functions 525
Creating a Driver Assignment 526
Editing a Driver Assignment 528
Global and Local Copies 529
Copying a Driver Assignment 531
Deleting a Driver Assignment 532

CHAPTER 21

CONSTANTS 535
Constants Functionality 538
Constants Screen Details 538
Creating a Constant 539
Editing a Constant 539
Copying a Constant 540
Deleting a Constant 542

CHAPTER 22

FILTER RULES 543
Filter Rules Options 544
Filter Rule Functionality 545
Filter Rule Tab 547
Filter Rule Functionality 548
Filter Rule Screen Details 548
Filter Rule Naming Information 549
Filter Rule Details 552
Filter Rule Available Variables, Functions and Constants 553
Filter Rule Report 554
Creating a Filter Rule 555
Editing a Filter Rule 557
Copying a Filter Rule 558
Copying a Filter Rule 561
Deleting a Filter Rule 563
Adding a Filter Rule to a Variable 564

CHAPTER 23

CALLOUTS 569
Same Project P2P Callouts 570
Cross Project Callouts 571
Using P2P Callouts 572
P2P Mapping Exceptions 573
Creating a P2P Exception Mapping 574
Map P2P Callouts 575
P2P Mapping Exception Screen Functionality 578
Adding an Exception 579
Creating a P2P Callout Step 580
Using P2L Callouts 583
Creating a P2L Exception mapping 584
Map P2L Callouts 585
To Create a P2L Callout Step 589
Removing a Program Mapping 590
Copying a Mapping 591
Copied or Versioned Programs 592
Implementing Callouts 592

CHAPTER 24

WORKING WITH REVISIONS 593
Creating an Element Revision 593
Changing the Active Revision of an Element 595
Deleting an Element Revision 596

CHAPTER 25

SEQUENCING 598
Sequencing Tab 599
Sequencing Tab Functions 600
Sequencing Algorithm/Driver Assignment Listings 602
Creating a Sequence 603
Multiple Selection Add or Remove 604
Changing Sequence Order 606

CHAPTER 26

OUTPUT MAPPING 608
Output Mapping Tab 609
Output Mapping Functionality 610
Output Mapping Screen Details 610
Creating a New Group 611
Editing Group Options 613
Edit Output Group Mapping Tab 614
Edit Output Group Mapping Tab Functionality 614
Edit Output Group Mapping Tab Screen Details 615
Edit Output Group Mapping Tab – Available Variables 616
Edit Output Group Mapping Tab – Used Variables 616
Editing an Output Mapping 617
Multiple Selection Add or Remove 619
Changing Output Order 621
Deleting a Group 622
Rating Worksheet 623
Worksheet Relationships 623
Creating a Worksheet 624
Editing a Worksheet 627
Worksheet Functions 628
Worksheet Screen Details 628
Algorithms 629
Steps 630
Reports 631
Adding Selections to a Worksheet 632
Worksheet Output XML 633
Full Dependency Report 634
Enabling/Disabling a Worksheet 635
Editing a Worksheet Name Using Options 636

Copying a Worksheet 637
Deleting a Worksheet 637
Mapped Worksheets 638
Enabling Worksheets in Test Case Editor 640

CHAPTER 27

INPUT AND OUTPUT MAPPING 642
Preparation for Mapping an Input File 643
Mapping Inputs Process 644
Custom XML Mapping Functionality 647
Custom XML Mapping Screen Details 647
Custom XML Mapping Work Areas 647
Importing Custom XML Using Choose Template 650
Importing Custom XML Using Copy/Paste 652
Creating a New Input Mapping 653
Save Mapping 656
View Mapping Xsl 658
Test Mapping Xsl 659
Import Insbridge Xsl 660
Upload Custom Input Mapping 661
Creating a New Output Mapping 662
Mapping Outputs 663
Importing Custom XML Using Choose Template 664
Importing Custom XML Using Copy/Paste 665
Creating an Output Mapping 666
Save Output Mapping 667
Rating a File with a Mapping 667
Using a Style Sheet Not Created in RateManager 668

CHAPTER 28

PACKAGING 669
Creating a Package 669
Versioning Package 673
Creating a Versioning Package: Utilizing Releases 673
Creating a Versioning Package: Not Allowing for Override 676
Resolving Packaging Errors 676

CHAPTER 29

TEST PROGRAM 678
Test Case Tab 679
Functionality 680
Creating a Test Case in Grid View 683
Creating a Test Case in XML View 686
Debug Tab 688
Hyperlinks to RateManager Elements 692

Opening an Existing Test Case 699

CHAPTER 30

SYSTEM REPORTS 703
Program Difference Report 705
Program Version Report 711
Inputs XML 714
Program Inputs XML 715
View Dependency Report 716
Full Dependency Report 717
View Dependency Report All Algorithm Above 718
View History 719
View PDF Report 720
View Quick Report 722
View Report 724
Table Difference Report 725
View Where Used 729

CHAPTER 31

COMMON FUNCTIONALITY 730
Locking 730
In-Use Locks 731
Program Version Locks 734
Locking a Program Version 734
Unlocking a Program Version 736
Template Locks 737
Notes 738
Adding Notes 739
Screen Options 741
Editing Notes 745
Deleting Notes 746
Change Control 747
Adding Change Control 747
Deleting Change Control 749
Custom Values 750
Adding a New Custom Value 750

SUPPORT

CONTACTING SUPPORT 751

APPENDIX A

INTERNET EXPLORER SETTINGS 752
How Security Levels Affect RateManager 752
Adding a Trusted Site 753
Microsoft Edge 756

Clearing Temp Files 758

APPENDIX B

COUNTRY CODES 760

GLOSSARY

GLOSSARY TERMS 767

INDEX

INDEX 773

LIST OF FIGURES

FIGURE 1 PROGRAM CREATION PROCESS 30
FIGURE 2 LOGIN SCREEN 40
FIGURE 3 LOGGING OUT 41
FIGURE 4 PROGRAM STRUCTURE IN RELEASE 4.9 46
FIGURE 5 PROGRAM STRUCTURE IN RELEASE 5.0 47
FIGURE 6 EXPANDED PROGRAM STRUCTURE IN 5.0 48
FIGURE 7 PROGRAM ELEMENT REVISIONS 56
FIGURE 8 HOME TAB 57
FIGURE 9 PREFERENCES SCREEN 59
FIGURE 10 UPDATED PREFERENCES 60
FIGURE 11 CHANGE PASSWORD SCREEN 61
FIGURE 12 PROGRAM EXPLORER WITHOUT CONTENT 62
FIGURE 13 PROGRAM EXPLORER WITH CONTENT 68
FIGURE 14 PROGRAM EXPLORER SEARCH 70
FIGURE 15 SCENARIO USER VIEW 77
FIGURE 16 SCENARIO PROGRAM SUBVERSION 78
FIGURE 17 SCENARIO TABLE SUBREVISIONS 78
FIGURE 18 SCENARIOS UPDATE REQUIRED 79
FIGURE 19 SUCCESSFUL SCENARIO CONVERSION 80
FIGURE 20 GROUP RIGHTS FOR SCENARIO MANAGERS 81
FIGURE 21 GROUP RIGHTS FOR SCENARIO USERS 82
FIGURE 22 RIGHT CLICK MENU FOR CREATING SCENARIOS 84
FIGURE 23 ADDING PROGRAM TO A SCENARIO 85
FIGURE 24 LOCKED PROGRAM UNAVAILABE TO A SCENARIO 86
FIGURE 25 REMOVING A PROGRAM FROM A SCENARIO 87
FIGURE 26 ADDING A TABLE TO A SCENARIO 88
FIGURE 27 REMOVING A TABLE FROM A SCENARIO 89
FIGURE 28 REMOVING A TABLE FROM A SCENARIO 90
FIGURE 29 USING VIEW WHERE USED TO CHECK TABLE STATUS 91
FIGURE 30 EDITING A SCENARIO 92
FIGURE 31 EDITING A SCENARIO NAME 93
FIGURE 32 PUBLISH OPTION 94
FIGURE 33 PROGRAM COMPARE REPORT 95
FIGURE 34 PUBLISHING OPTIONS FOR A SCENARIO 96
FIGURE 35 COMPLETED PUBLISH PROGRAM 97
FIGURE 36 UNLOCK CONFIRMATION MESSAGE 98
FIGURE 37 DELETING A SCENARIO 99
FIGURE 38 DELETING A SCENARIO MESSAGE 99
FIGURE 39 PROGRAM EXPLORER FOR A SCENARIO USER 101
FIGURE 40 PROGRAM DETAILS FOR A SCENARIO USER 102
FIGURE 41 EDITING A TABLE FOR A SCENARIO USER 103
FIGURE 42 CREATING A NEW OTHER DOMAIN TYPE 106
FIGURE 43 EDITING A DOMAIN TYPE 107
FIGURE 44 DELETING A DOMAIN TYPE 108
FIGURE 45 OTHER DOMAIN TYPE RIGHT CLICK MENU 109
FIGURE 46 RIGHT CLICK FOR NEW DOMAIN 110
FIGURE 47 NEW DOMAIN 110
FIGURE 48 PROGRAM EXPLORER WITH NEW DOMAIN 111
FIGURE 49 EDITING A DOMAIN 111
FIGURE 50 EDITING A DOMAIN ON THE EDIT DOMAIN SCREEN 112
FIGURE 51 CLEARING A DOMAIN 113
FIGURE 52 FIRST CONFIRMATION TO CLEAR THE DOMAIN 113
FIGURE 53 SECOND CONFIRMATION TO CLEAR THE DOMAIN 114
FIGURE 54 DELETING A DOMAIN 115
FIGURE 55 FIRST CONFIRMATION TO DELETE THE DOMAIN 115
FIGURE 56 SECOND CONFIRMATION TO DELETE THE DOMAIN 116
FIGURE 57 DOMAIN RIGHT CLICK MENU 116
FIGURE 58 DOMAIN RIGHT CLICK MENU FOR NEW PROJECT 118
FIGURE 59 CREATING A PROJECT 118

FIGURE 60 NO ACCESS TO PRODUCTS 119
FIGURE 61 SELECTING PRODUCTS RIGHT CLICK MENU 120
FIGURE 62 EDITING A PROJECT 121
FIGURE 63 FIRST CONFIRMATION TO CLEAR THE PROJECT 122
FIGURE 64 CLEARED PROJECT 122
FIGURE 65 MOVING A PROJECT 123
FIGURE 66 MOVE PROJECT OPTIONS 124
FIGURE 67 FINAL LOCATION FOR MOVED PROJECT 124
FIGURE 68 DELETING A PROJECT 125
FIGURE 69 GLOBAL VERSIONING REPORT NOTIFICATION 127
FIGURE 70 GLOBAL VERSIONING REPORT 128
FIGURE 71 SELECTING CATEGORY VISIBILITY 129
FIGURE 72 MANAGE FOLDERS 132
FIGURE 73 RIGHT CLICK MENU FOR FOLDERS 133
FIGURE 74 RIGHT CLICK MENU FOR PROGRAM VERSION FOLDERS 134
FIGURE 75 CREATING NEW FOLDER 135
FIGURE 76 RENAMING A FOLDER 136
FIGURE 77 DELETING A FOLDER 137
FIGURE 78 CONFIRMING A FOLDER DELETE 137
FIGURE 79 FOLDER RIGHT CLICK MENU 138
FIGURE 80 FOLDER OPTIONS FOUND ON THE RIGHT CLICK MENU OF PROGRAMS 139
FIGURE 81 EDIT FOLDER 140
FIGURE 82 FOLDER OPTIONS FOUND ON THE RIGHT CLICK MENU OF PROGRAM DETAILS 141
FIGURE 83 MOVING A FOLDER 142
FIGURE 84 CREATING NEW PROGRAM VERSION FOLDER 143
FIGURE 85 RENAMING A FOLDER 144
FIGURE 86 DELETING A PROGRAM VERSION FOLDER 145
FIGURE 87 PROGRAMS ON PROGRAM EXPLORER 146
FIGURE 88 CREATING A NEW PROGRAM 153
FIGURE 89 SELECTING DESTINATION FOR MOVING A PROGRAM 154
FIGURE 90 EDITING A PROGRAM 155
FIGURE 91 DELETE WARNING MESSAGE 156
FIGURE 92 CONFIRMING DELETE 156
FIGURE 93 PROGRAM VERSIONS IN PROGRAM EXPLORER 157
FIGURE 94 ADDING A NEW VERSION 159
FIGURE 95 COPY PROGRAM VERSION 160
FIGURE 96 DELETE WARNING MESSAGE 161
FIGURE 97 CONFIRMING DELETE 161
FIGURE 98 EDITING A PROGRAM VERSION 162
FIGURE 99 ADDING VERSIONING CRITERIA TO A PROGRAM 164
FIGURE 100 RIGHT CLICK MENU FOR THE SETTING VERSIONING CRITERIA 166
FIGURE 101 EDITING A PROGRAM VERSION 168
FIGURE 102 PROGRAM EXPLORER WITH VERSION NAMES AND VERSION FOLDERS 171
FIGURE 103 CREATING A PROGRAM VERSION NAME 172
FIGURE 104 ENTERING A PROGRAM VERSION NAME 173
FIGURE 105 ASSIGNING A PROGRAM VERSION FOLDER 175
FIGURE 106 ASSIGNING A PROGRAM TO A PROGRAM VERSION FOLDER 176
FIGURE 107 VIEWING PROGRAM VERSION FOLDERS CONTENTS 177
FIGURE 108 UNASSIGNING A PROGRAM VERSION 177
FIGURE 109 PROGRAM DETAILS 180
FIGURE 110 NAVIGATING TO PROGRAM DETAILS 181
FIGURE 111 ELEMENT INFORMATION WHEN SCROLLING 195
FIGURE 112 ELEMENT DETAILS 195
FIGURE 113 SELECTING UNIVERSALS FOR THE PROJECT 198
FIGURE 114 UNIVERSAL TAB 199
FIGURE 115 SELECTING TO MANAGE PROJECT RELATIONSHIPS 200
FIGURE 116 EDITING CROSS PROJECT RELATIONSHIPS 201
FIGURE 117 GRANTING ACCESS TO A PROJECT 202
FIGURE 118 COMPLETED MANAGE PROJECT RELATIONSHIP 203
FIGURE 119 AVAILABLE DOMAINS FOR PROJECT RELATIONSHIPS 204
FIGURE 120 DOMAINS FOR PROJECT RELATIONSHIPS 204
FIGURE 121 AVAILABLE PROJECTS FOR SELECTION 205

FIGURE 122 SELECTED PROJECT 205
FIGURE 123 EDITING ACCESS FOR A PROJECT RELATIONSHIP 206
FIGURE 124 DENYING ACCESS FOR A PROJECT 207
FIGURE 125 UPDATED MANAGE PROJECT RELATIONSHIP 207
FIGURE 126 SELECTING GLOBALS FOR THE 209
FIGURE 127 GLOBAL TAB 210
FIGURE 128 CATEGORIES OPTION ON THE UNIVERSALS TAB 212
FIGURE 129 CATEGORIES SCREEN 213
FIGURE 130 CREATING NEW CATEGORY 216
FIGURE 131 EDITING A CATEGORY 217
FIGURE 132 DELETING A CATEGORY 219
FIGURE 133 SELECTING FIELDS 220
FIGURE 134 FIELD LISTING 221
FIGURE 135 NEW FIELD VARIABLE WINDOW 224
FIGURE 136 CUSTOM XML ID ERROR 225
FIGURE 137 ADDING A FIELD VARIABLE 226
FIGURE 138 EDITING FIELD VARIABLE 227
FIGURE 139 COPYING UNIVERSAL FIELD VARIABLE 228
FIGURE 140 DELETING A FIELD 229
FIGURE 141 DEPRECATE FIELD CONFIRMATION 229
FIGURE 142 RESTORE FIELD CONFIRMATION 230
FIGURE 143 DELETING A FIELD 231
FIGURE 144 SELECTING VALIDATIONS 232
FIGURE 145 VALIDATIONS LISTING 233
FIGURE 146 CREATING A LIST OF VALUES 239
FIGURE 147 ENTERING DETAILS FOR LIST OF VALUES 240
FIGURE 148 EDITING THE VALUE 240
FIGURE 149 EDITING A LIST OF VALUES 241
FIGURE 150 ADDING A LIST OF VALUES TO A FIELD 242
FIGURE 151 SELECTING A LIST OF VALUES 243
FIGURE 152 FIELD WITH LIST OF VALUES ADDED 243
FIGURE 153 TABLE VARIABLE WITH LIST OF VALUES FIELD 244
FIGURE 154 IMPORTING DATA WITH LIST OF VALUES FIELD 245
FIGURE 155 EDITING DATA WITH A LIST OF VALUES FIELD 246
FIGURE 156 VALID VALUES FOR TABLE DATA 247
FIGURE 157 SELECTING TO VIEW TABLE DATE BY LABEL 247
FIGURE 158 LIST OF VALUES IN TEST CASE EDITOR 248
FIGURE 159 EDITING A LIST OF VALUES DATA TABLE 249
FIGURE 160 EDITING A ROW ON THE LIST OF VALUES DATA TABLE 250
FIGURE 161 COPYING LIST OF VALUES 251
FIGURE 162 SETTING THE ACTIVE REVISION OF A LIST OF VALUES 252
FIGURE 163 DELETING A LIST OF VALUES 253
FIGURE 164 SELECTING RESULT VARIABLES 254
FIGURE 165 RESULT VARIABLES TAB 255
FIGURE 166 NEW RESULT VARIABLE 257
FIGURE 167 ADDING A RESULT VARIABLE 258
FIGURE 168 EDITING RESULT VARIABLE 259
FIGURE 169 COPYING RESULT VARIABLE 260
FIGURE 170 DELETING A RESULT VARIABLE 261
FIGURE 171 RESULT VARIABLE ERROR MESSAGE 261
FIGURE 172 TABLE VARIABLE NAVIGATION 263
FIGURE 173 TABLE VARIABLE TAB 265
FIGURE 174 TABLE LINKED VARIABLE FIELD 268
FIGURE 175 TABLE VARIABLE INFORMATION FIELD 269
FIGURE 176 TABLE AVAILABLE VARIABLE FIELD 271
FIGURE 177 TABLE VARIABLE SELECTION CRITERIA FIELD 272
FIGURE 178 MASK VALUE DIALOGUE BOX 274
FIGURE 179 CREATING A TABLE VARIABLE 277
FIGURE 180 CREATING A LINKED TABLE VARIABLE 279
FIGURE 181 UNIVERSAL TABLE VARIABLE COPIED TO A LOCAL LEVEL 281
FIGURE 182 COPYING TABLE VARIABLE AT THE GLOBAL LEVEL 282
FIGURE 183 COPYING TABLE VARIABLE AT THE LOCAL LEVEL 283

FIGURE 184 COPYING A GLOBAL TABLE VARIABLE....285
FIGURE 185 DELETING A VARIABLE....286
FIGURE 186 ERROR WHEN DELETING A VARIABLE....286
FIGURE 187 SELECTING DATA FILE....290
FIGURE 188 IMPORT TABLE DATA....291
FIGURE 189 CONFIRMING PREVIEW OF DATA....291
FIGURE 190 PREVIEWING DATA FILE WITH AUTOMATICALLY MATCHED COLUMNS....292
FIGURE 191 PREVIEWING DATA FILE WITH MANUAL MATCHING REQUIRED....293
FIGURE 192 CONFIRMING DELETE ALL DATA....294
FIGURE 193 SUCCESSFUL IMPORT MESSAGE....295
FIGURE 194 IMPORTING TABLE DATA FROM AN EXCEL FILE....296
FIGURE 195 IMPORT EXCEL FILE OPTIONS....297
FIGURE 196 SELECTING DATA FILE....298
FIGURE 197 SELECTING A FILE FOR IMPORT....299
FIGURE 198 IMPORT TABLE DATA....299
FIGURE 199 CONFIRMING PREVIEW OF DATA....300
FIGURE 200 PREVIEWING DATA FILE WITH AUTOMATICALLY MATCHED COLUMNS....301
FIGURE 201 PREVIEWING DATA FILE WITH MANUAL MATCHING REQUIRED....302
FIGURE 202 IMPORT FILE JOB OPTIONS....303
FIGURE 203 SELECTING A JOB FOR IMPORT....304
FIGURE 204 TABLE JOB MANAGEMENT....304
FIGURE 205 SUCCESSFUL IMPORT TABLE JOB....305
FIGURE 206 PREVIEWING DATA FILE WITH A DATE MASK....306
FIGURE 207 IMPORTED TABLE DATA....307
FIGURE 208 SELECTING A TABLE FOR EXPORT....309
FIGURE 209 EXPORT DATA INCLUDING WILDCARD NULL VALUES....310
FIGURE 210 INCLUDE ASTERISKS FOR WILDCARD NULL VALUES....311
FIGURE 211 SAVING A TABLE....312
FIGURE 212 EXPORT DATA FILE....314
FIGURE 213 OPTIONS FOR EXPORTING TABLE DATA....315
FIGURE 214 SELECTING A JOB....316
FIGURE 215 JOBS LISTED ON TABLE JOB MANAGEMENT....316
FIGURE 216 OVERWRITING A JOB....317
FIGURE 217 EXPORTED TABLE IN EXCEL....318
FIGURE 218 CRITERIA DEFINITIONS FOR EXPORTED TABLE VARIABLES....319
FIGURE 219 TABLE IDENTIFIERS AND RESULTS DEFINITIONS....320
FIGURE 220 TABLE DATA DEFINITIONS....320
FIGURE 221 EDITING TABLE DATA....321
FIGURE 222 NAVIGATING EDIT DATA SCREEN....322
FIGURE 223 EDITING TABLE DATA....324
FIGURE 224 EDIT TABLE VARIABLE DATA....324
FIGURE 225 EDITING TABLE DATA....325
FIGURE 226 INSERTING ROWS ON TABLE VARIABLE DATA....326
FIGURE 227 INSERTING ROW IN TABLE....326
FIGURE 228 DELETING ROWS FROM TABLE DATA....327
FIGURE 229 DELETING ALL ROWS IN TABLE....327
FIGURE 230 CONFIRMING ALL ROWS DELETE....328
FIGURE 231 COPYING ROWS OF TABLE DATA....329
FIGURE 232 PASTING ROWS OF DATA IN A TABLE....330
FIGURE 233 FILTERED TABLE DATA....331
FIGURE 234 SUSPENDING A FILTER....332
FIGURE 235 SORTED AND FILTERED TABLE DATA....333
FIGURE 236 FILTERED DATA FOR EXPORT....334
FIGURE 237 TABLE DATA WITH A LIST OF VALUES....336
FIGURE 238 VALID VALUES FOR TABLE DATA....337
FIGURE 239 SELECTING TO VIEW TABLE DATE BY LABEL....337
FIGURE 240 DATA SOURCE ON PROGRAM DETAILS....338
FIGURE 241 STANDARD CRITERIA....339
FIGURE 242 GROUPED CRITERIA....339
FIGURE 243 DATA SOURCES GROUPING....340
FIGURE 244 NAVIGATION TO NEW DATA SOURCES....341
FIGURE 245 CREATING A DATA SOURCE....341

FIGURE 246 NEW COLUMN FOR DATA SOURCE 342
FIGURE 247 EDITING DATA SOURCE DATA 343
FIGURE 248 NAVIGATING EDITING DATA SCREEN 344
FIGURE 249 EDITING DATA SOURCES DATA 346
FIGURE 250 EDITING DATA SOURCES 347
FIGURE 251 SAVING DATA ON EDITING A DATA SOURCE 348
FIGURE 252 DELETING ROWS FROM DATA SOURCE 349
FIGURE 253 EDITING DATA SOURCES DATA 350
FIGURE 254 INSERTING ROWS ON DATA SOURCE 351
FIGURE 255 DELETING ROWS FROM DATA SOURCE 352
FIGURE 256 CONFIRMING ALL ROWS DELETE 353
FIGURE 257 FILTERED DATA SOURCE DATA 354
FIGURE 258 SUSPENDING A FILTER 355
FIGURE 259 SORTED DATA SOURCE DATA 356
FIGURE 260 FILTERED DATA FOR EXPORT 358
FIGURE 261 DELETING A DATA SOURCE 359
FIGURE 262 DATA SOURCE ON OUTPUT MAPPING 360
FIGURE 263 ENTERING A NEW STEP 363
FIGURE 264 COPY VARIABLE MENU 365
FIGURE 265 SUCCESSFUL COPY 365
FIGURE 266 PASTING STEPS 366
FIGURE 267 SUCCESSFUL PASTING OF STEP 366
FIGURE 268 PASTED STEP HIGHLIGHTED 367
FIGURE 269 CLIPBOARD CONTENTS 368
FIGURE 270 SUCCESSFUL CLEARING OF CLIPBOARD 368
FIGURE 271 SETTING THE NEXT STEP 369
FIGURE 272 RESULTS OF STEP 370
FIGURE 273 EDITING STEPS 372
FIGURE 274 SETTING THE STEP OPERATOR 373
FIGURE 275 COMPARISON OPERATORS TABLE 373
FIGURE 276 SETTING THE ROUNDING OPTION 374
FIGURE 277 WHERE CLAUSE CATEGORY STRUCTURE 377
FIGURE 278 PARENTHESES IN THE ALGORITHM REPORT 381
FIGURE 279 ADDING A WHERE CLAUSE 382
FIGURE 280 COMPLETING THE WHERE CLAUSE 382
FIGURE 281 DELETING A WHERE CLAUSE 383
FIGURE 282 ADDING CONDITIONS TO A WHERE CLAUSE 383
FIGURE 283 EDITING A WHERE CLAUSE 384
FIGURE 284 ADDING AN OPEN PARENTHESES 385
FIGURE 285 ADDING A CLOSE PARENTHESES 385
FIGURE 286 MOVING STEPS 387
FIGURE 287 CHANGE STEP ORDER 388
FIGURE 288 DELETING A STEP TERM 389
FIGURE 289 DELETE STEP TERM WARNING 389
FIGURE 290 DELETING A STEP 390
FIGURE 291 DELETE STEP WARNING 390
FIGURE 292 GET CATEGORY STEP 391
FIGURE 293 SET CATEGORY ITEM STEP 394
FIGURE 294 GET RANKED CATEGORY 396
FIGURE 295 SETTING RANKED CATEGORY ITEM 397
FIGURE 296 COUNT ACROSS CATEGORY STEP 397
FIGURE 297 ISALPHA STEP 400
FIGURE 298 ISDATE STEP 401
FIGURE 299 ISNUMERIC STEP 401
FIGURE 300 DATE ADDITION 402
FIGURE 301 DATE DIFFERENCE 402
FIGURE 302 DIFFERENCE SIGN SELECTION 403
FIGURE 303 CREATING A SIMPLE IF STATEMENT 406
FIGURE 304 INSERTING CONDITIONS IN AN IF STATEMENT 407
FIGURE 305 THEN AND ELSE SEGMENTS OF AN IF STATEMENT 407
FIGURE 306 EDITING A MULTIPLE IF STATEMENT 408
FIGURE 307 MASK STEP 409

FIGURE 308 AVAILABLE MASK OPTIONS TABLE....409
FIGURE 309 MASK OPTION EXAMPLES TABLE....409
FIGURE 310 ARITHMETIC STEP....410
FIGURE 311 SETTING AN ARITHMETIC STEP 2....411
FIGURE 312 ABSOLUTE VALUE....412
FIGURE 313 POWER STEP....413
FIGURE 314 LOG STEP....413
FIGURE 315 LOG10 STEP....413
FIGURE 316 NATURAL LOG STEP....414
FIGURE 317 EXPONENT STEP....414
FIGURE 318 RANDOM NUMBER STEP....415
FIGURE 319 FACTORIAL STEP....415
FIGURE 320 SQUARE ROOT STEP....415
FIGURE 321 CEILING STEP....416
FIGURE 322 FLOOR STEP....416
FIGURE 323 EVEN STEP....417
FIGURE 324 ODD STEP....417
FIGURE 325 COSINE STEP....418
FIGURE 326 COSH STEP....418
FIGURE 327 ACOS STEP....419
FIGURE 328 ACOSH STEP....419
FIGURE 329 SINE STEP....419
FIGURE 330 SINH STEP....420
FIGURE 331 ASIN STEP....420
FIGURE 332 ASINH STEP....421
FIGURE 333 TANGENT STEP....421
FIGURE 334 TANH STEP....421
FIGURE 335 ATAN STEP....422
FIGURE 336 ATANH STEP....422
FIGURE 337 DEGREES STEP....423
FIGURE 338 RADIANS STEP....423
FIGURE 339 LOG GAMMA STEP....424
FIGURE 340 INCOMPLETE GAMMA STEP....424
FIGURE 341 GAMMA STEP....425
FIGURE 342 CLEAR RANKING STEP....426
FIGURE 343 RANK ACROSS CATEGORY STEP....426
FIGURE 344 GET LENGTH....427
FIGURE 345 SET MESSAGE....427
FIGURE 346 STRING ADDITION STEP....427
FIGURE 347 SUM ACROSS CATEGORY STEP....428
FIGURE 348 MULTIPLE ACROSS CATEGORY STEP....431
FIGURE 349 MINIMUM/MAXIMUM STEP....431
FIGURE 350 CREATING A MAXIMUM VALUE STEP....432
FIGURE 351 SETTING UNDERWRITING TO FAIL STEP....432
FIGURE 352 SET UNDERWRITING TO FAIL AND SET MESSAGE STEPS....433
FIGURE 353 SET PRINCIPAL OPERATOR VARIABLE....434
FIGURE 354 RE-CALCULATE VEHICLE USAGE STEP....434
FIGURE 355 ENTERING A DATA SOURCE QUERY....436
FIGURE 356 SELECTING DATA SOURCES IN A LOCAL ALGORITHM....437
FIGURE 357 COMPLETING A DATA QUERY ALGORITHM....438
FIGURE 358 CALCULATED VARIABLE NAVIGATION....440
FIGURE 359 CALCULATED VARIABLE TAB....442
FIGURE 360 CALCULATED VARIABLE INFORMATION FIELD....444
FIGURE 361 CALCULATED VARIABLE SELECTION CRITERIA FIELD....447
FIGURE 362 AVAILABLE VARIABLES, FUNCTIONS AND CONSTANTS FIELD....448
FIGURE 363 CALCULATED VARIABLE REPORT....450
FIGURE 364 CREATING A NEW CALCULATED VARIABLE....451
FIGURE 365 COPYING CALCULATED VARIABLE AT THE LOCAL LEVEL....455
FIGURE 366 COPYING CALCULATED VARIABLE AT THE GLOBAL LEVEL....456
FIGURE 367 COPY VARIABLE MENU....457
FIGURE 368 COPY CALCULATED VARIABLE....457
FIGURE 369 DELETING A VARIABLE....459

FIGURE 370 ERROR MESSAGE FOR DELETING A VARIABLE....459
FIGURE 371 RESOLVING DEPENDENT VARIABLES....460
FIGURE 372 ALGORITHMS NAVIGATION....464
FIGURE 373 ALGORITHM TAB....466
FIGURE 374 ALGORITHM INFORMATION FIELD....468
FIGURE 375 ALGORITHM STEP DETAILS....470
FIGURE 376 AVAILABLE VARIABLES, FUNCTION AND CONSTANTS FIELD....471
FIGURE 377 ALGORITHM REPORT....473
FIGURE 378 CREATING A NEW ALGORITHM....474
FIGURE 379 ADDING AN ADVANCED OPTION TO A RATING ALGORITHM....476
FIGURE 380 ALGORITHM ADVANCED OPTIONS....477
FIGURE 381 ADVANCED OPTIONS ON....477
FIGURE 382 ALGORITHM WITH ALGORITHM LOOPING APPLIED....478
FIGURE 383 CREATING A CATEGORY INSTANCE....479
FIGURE 384 SELECTING A CATEGORY FOR DYNAMIC CATEGORY INSTANCE....480
FIGURE 385 SELECTING THE CATEGORY FOR DYNAMIC CATEGORY INSTANCE....481
FIGURE 386 ADVANCED OPTIONS DO WHILE LOOPING....482
FIGURE 387 ADDING A CREATE CATEGORIES INSTANCE MAPPING....483
FIGURE 388 USING THE DYNAMIC CATEGORY INSTANCE INDICATOR....484
FIGURE 389 SELECTING MAP CALLOUTS....485
FIGURE 390 CREATING A NEW MAPPING....486
FIGURE 391 SELECTING THE CATEGORY....486
FIGURE 392 VIEW/EDIT MAPPINGS....487
FIGURE 393 SELECTING A ADD MAPPINGS....487
FIGURE 394 ADDING A MAPPING....488
FIGURE 395 ADDING MAPPINGS TO CREATE CATEGORY INSTANCE....489
FIGURE 396 MESSAGE TEMPLATES....490
FIGURE 397 COPYING AN ALGORITHM AT THE GLOBAL LEVEL....493
FIGURE 398 COPYING AN ALGORITHM AT THE LOCAL LEVEL....493
FIGURE 399 COPY ALGORITHM MENU....494
FIGURE 400 COPY ALGORITHM....495
FIGURE 401 DELETING AN ALGORITHM....496
FIGURE 402 DRIVER ASSIGNMENT NAVIGATION....501
FIGURE 403 DRIVER ASSIGNMENT TAB....503
FIGURE 404 DRIVER ASSIGNMENT INFORMATION FIELD....506
FIGURE 405 ALGORITHM STEP DETAIL....508
FIGURE 406 AVAILABLE VARIABLES, FUNCTION AND CONSTANTS FIELD....510
FIGURE 407 ALGORITHM REPORT....512
FIGURE 408 FUNCTIONS MENU FOR DRIVER ASSIGNMENT SCENARIO....513
FIGURE 409 FLAGGING FUNCTIONS....514
FIGURE 410 RANKING FUNCTIONS....518
FIGURE 411 ADDING AN ASSIGNMENT FUNCTION....521
FIGURE 412 CREATING A DRIVER ASSIGNMENT....526
FIGURE 413 COPYING DRIVER ASSIGNMENT AT THE GLOBAL LEVEL....529
FIGURE 414 COPYING DRIVER ASSIGNMENT AT THE LOCAL LEVEL....530
FIGURE 415 COPY DRIVER ASSIGNMENT MENU....531
FIGURE 416 COPY DRIVER ASSIGNMENT....532
FIGURE 417 DELETING A DRIVER ASSIGNMENT....533
FIGURE 418 DRIVER ASSIGNMENT DELETE ERROR MESSAGE....534
FIGURE 419 UNIVERSAL USER DEFINED CONSTANTS....535
FIGURE 420 CONSTANTS LISTING ON PROGRAM DETAILS....537
FIGURE 421 UNIVERSAL CONSTANTS VARIABLE....539
FIGURE 422 EDITING A UNIVERSAL LEVEL CONSTANT....540
FIGURE 423 COPYING GLOBAL CONSTANT VARIABLE....541
FIGURE 424 DELETING A CONSTANT....542
FIGURE 425 FILTER RULE NAVIGATION....545
FIGURE 426 FILTER RULE TAB....547
FIGURE 427 FILTER RULES NAMING INFORMATION....549
FIGURE 428 FILTER RULES VARIABLE DETAILS....552
FIGURE 429 AVAILABLE VARIABLES, FUNCTIONS AND CONSTANTS....553
FIGURE 430 FILTER RULES REPORT....554
FIGURE 431 CREATING A NEW FILTER RULE....555

FIGURE 432 SELECTING A TRUE OR FALSE....556
FIGURE 433 COPYING FILTER RULES AT THE GLOBAL LEVEL....559
FIGURE 434 COPYING FILTER RULE AT THE LOCAL LEVEL....560
FIGURE 435 COPY FILTER RULE MENU....561
FIGURE 436 COPY FILTER RULE SCREEN....561
FIGURE 437 DELETING A FILTER RULE....563
FIGURE 438 ERROR MESSAGE FOR DELETING A FILTER RULE....563
FIGURE 439 ADDING A FILTER RULE TO A VARIABLE....564
FIGURE 440 SELECT FILTER RULE BY CATEGORY....566
FIGURE 441 UNIVERSAL TABLE VARIABLE WITH A FILTER RULE....568
FIGURE 442 CALLOUT ALGORITHM....569
FIGURE 443 SAME PROJECT P2P....570
FIGURE 444 CROSS PROJECT P2P....571
FIGURE 445 NAVIGATING TO CALLOUTS....574
FIGURE 446 MAP CALLOUT EXCEPTIONS....575
FIGURE 447 CREATING AN EXCEPTION MAPPING....576
FIGURE 448 EDIT FIELDS....577
FIGURE 449 ADDING EXCEPTIONS....578
FIGURE 450 EXCEPTION MAPPING....579
FIGURE 451 SELECTING FUNCTIONS – PROGRAM CALLOUTS....580
FIGURE 452 SELECTING A PROGRAM....581
FIGURE 453 CALLOUT LISTED AS A STEP....582
FIGURE 454 NAVIGATING TO CALLOUTS....584
FIGURE 455 MAP CALLOUT EXCEPTIONS....585
FIGURE 456 CREATING A LIBRARY CALLOUT MAPPING....586
FIGURE 457 EDIT FIELDS....587
FIGURE 458 ADDING EXCEPTIONS....587
FIGURE 459 SELECTING A P2L CALLOUT....589
FIGURE 460 P2L LISTED AS A STEP....590
FIGURE 461 COPYING A MAPPING....591
FIGURE 462 COPY CALLOUT MAPPING....591
FIGURE 463 SELECTING A VARIABLE TO REVISE....593
FIGURE 464 REVISING A VARIABLE....594
FIGURE 465 SELECTING AN ACTIVE REVISION....595
FIGURE 466 CONFIRMATION MESSAGE....596
FIGURE 467 DELETING A REVISION....597
FIGURE 468 UNABLE TO DELETE REVISION DUE TO LINKED VARIABLE....597
FIGURE 469 NAVIGATING TO SEQUENCING TAB....598
FIGURE 470 SEQUENCING TAB....599
FIGURE 471 SEQUENCE PAGING....601
FIGURE 472 COMPLETED SEQUENCE....603
FIGURE 473 MULTIPLE SELECTIONS TO CREATE A SEQUENCE....605
FIGURE 474 CHANGE SEQUENCE....606
FIGURE 475 CHANGING SEQUENCE ORDER USING DROP AND DRAG....607
FIGURE 476 NAVIGATING TO OUTPUT MAPPING TAB....608
FIGURE 477 OUTPUT MAPPING TAB....609
FIGURE 478 ENTERING A NEW OUTPUT GROUP....612
FIGURE 479 EDIT OUTPUT GROUP....613
FIGURE 480 EDITING AN OUTPUT MAPPING....614
FIGURE 481 AVAILABLE VARIABLES FOR OUTPUT MAPPING....616
FIGURE 482 VARIABLES USED IN OUTPUT MAPPING....617
FIGURE 483 EDITING AN OUTPUT GROUP MAPPING....618
FIGURE 484 MULTIPLE SELECTIONS TO EDIT AN OUTPUT MAPPING....620
FIGURE 485 MOVING OUTPUT MAPPING....621
FIGURE 486 OUTPUT CODES....621
FIGURE 487 DELETING AN OUTPUT GROUP MAPPING....622
FIGURE 488 CREATING A NEW WORKSHEET....624
FIGURE 489 NAMING A WORKSHEET....625
FIGURE 490 SELECTING A WORKSHEET TO EDIT....625
FIGURE 491 RIGHT CLICK MENU FOR WORKSHEETS....626
FIGURE 492 EDIT WORKSHEET SCREEN....627
FIGURE 493 EDIT WORKSHEET ALGORITHMS....629

FIGURE 494 EDIT WORKSHEET STEPS....630
FIGURE 495 EDIT WORKSHEET REPORT....631
FIGURE 496 SELECTING ALGORITHM STEPS FOR WORKSHEET....632
FIGURE 497 WORKSHEET XML VIEWER....633
FIGURE 498 XML WORKSHEET ERROR....633
FIGURE 499 WORKSHEET DEPENDENCY REPORT....634
FIGURE 500 WORKSHEET OPTIONS....635
FIGURE 501 ENABLING WORKSHEET....636
FIGURE 502 EDITING A WORKSHEET....636
FIGURE 503 COPYING A WORKSHEET....637
FIGURE 504 DELETING A WORKSHEET....637
FIGURE 505 SELECTING THE WORKSHEET FOR THE OUTPUT GROUP....638
FIGURE 506 MAPPED WORKSHEET TO AN OUTPUT GROUP....639
FIGURE 507 ENABLING WORKSHEETS IN PREFERENCES....640
FIGURE 508 WORKSHEET OPTION IN TEST CASE EDITOR....641
FIGURE 509 USING AN EXISTING XML FORMAT....642
FIGURE 510 CUSTOM XML MAPPING AT GLOBAL LEVEL....645
FIGURE 511 CUSTOM XML MAPPING AT THE PROGRAM VERSION LEVEL....646
FIGURE 512 CUSTOM XML WORK AREA....648
FIGURE 513 CATEGORY INPUTS WORK AREA....649
FIGURE 514 UPLOADING A CUSTOM XML TEMPLATE....650
FIGURE 515 IMPORTED CUSTOM XML TEMPLATE....651
FIGURE 516 IMPORT CUSTOM XML TEXT BOX....652
FIGURE 517 IMPORTED CUSTOM XML USING COPY AND PASTE....653
FIGURE 518 MAPPING CUSTOM XML....654
FIGURE 519 MAPPING SYSTEM CATEGORY: RATE....654
FIGURE 520 SELECTING CATEGORIES....655
FIGURE 521 SELECTING INPUTS....655
FIGURE 522 SAVE MAPPING....656
FIGURE 523 VIEW MAPPINGS XSL....658
FIGURE 524 TEST MAPPING XSL....659
FIGURE 525 OPEN MAPPING RIGHT CLICK MENU....660
FIGURE 526 UPLOAD CUSTOM INPUT MAPPING....661
FIGURE 527 NAVIGATING TO OUTPUT MAPPING....663
FIGURE 528 CREATING A CUSTOM OUTPUT MAPPING....663
FIGURE 529 MAPPING CUSTOM OUTPUT XML....664
FIGURE 530 UPLOADING A CUSTOM OUTPUT XML TEMPLATE....665
FIGURE 531 CREATING A PACKAGE....670
FIGURE 532 CREATING A PACKAGE RELEASE PACKAGING SET TO FALSE....672
FIGURE 533 CREATING A VERSIONING PACKAGE....673
FIGURE 534....674
FIGURE 535 GLOBAL PACKAGING SUCCESS MESSAGE....675
FIGURE 536 PACKAGE FAILURE....677
FIGURE 537 SELECTING TEST PROGRAM....678
FIGURE 538 TEST PROGRAM TAB....679
FIGURE 539 RIGHT CLICK MENU FOR TEST CASE....681
FIGURE 540 INPUTS FOR SELECTED CATEGORY....683
FIGURE 541 ENTERING INPUTS IN TEST PROGRAM....684
FIGURE 542 SAVING TEST CASE FILES....685
FIGURE 543 TEST CASE XML VIEW....686
FIGURE 544 XML VALIDATION....687
FIGURE 545 DEBUG SINGLE REPORT FORMAT....688
FIGURE 546 DEBUG REPORT IN SUMMARY FORMAT....689
FIGURE 547 DEBUG REPORT POP OUT WINDOW....689
FIGURE 548 VIEW DEBUG REPORTS....690
FIGURE 549 SELECT PROGRAM POPUP....691
FIGURE 550 HYPERLINKS IN DEBUG REPORT....692
FIGURE 551 EDITING AN ELEMENT FROM DEBUG REPORT....693
FIGURE 552 ADVANCED OPTIONS IN DEBUG....693
FIGURE 553 LOOPING DETAILS....694
FIGURE 554 VIEW DEBUG REPORTS XML....695
FIGURE 555 WORKSHEET OPTION IN TEST CASE EDITOR....696

FIGURE 556 SAMPLE WORKSHEET REPORT 696
FIGURE 557 MESSAGE TEMPLATES IN DEBUG REPORT 697
FIGURE 558 MESSAGE TEMPLATES IN XML DEBUG REPORT 697
FIGURE 559 SIGNED VALUES IN DEBUG REPORT 698
FIGURE 560 SIGNED VALUES FOR CATEGORY ITEMS 698
FIGURE 561 OPENING A TEST CASE 699
FIGURE 562 RIGHT CLICK MENU FOR OPENING TEST CASES 700
FIGURE 563 TEST CASE ERROR 701
FIGURE 564 OPENING A LOCAL TEST CASE 702
FIGURE 565 PROGRAM DIFFERENCE REPORT RIGHT CLICK MENU 706
FIGURE 566 SELECTING PROGRAMS VERSIONS TO COMPARE 707
FIGURE 567 COMPLETED PROGRAM DIFFERENCE REPORT 708
FIGURE 568 PROGRAM DIFFERENCE REPORT 709
FIGURE 569 PROGRAM VERSION REPORT RIGHT CLICK MENU 711
FIGURE 570 PROGRAM VERSION REPORT 712
FIGURE 571 INPUTS XML 714
FIGURE 572 PROGRAM INPUTS XML 715
FIGURE 573 DEPENDENCY REPORT 716
FIGURE 574 FULL DEPENDENCY REPORT 717
FIGURE 575 VIEW DEPENDENCY REPORT ALL ALGORITHM ABOVE 718
FIGURE 576 VIEW HISTORY 719
FIGURE 577 VIEW PDF REPORT 720
FIGURE 578 VIEW QUICK REPORT ALL DATA 722
FIGURE 579 VIEW QUICK REPORT 723
FIGURE 580 VIEW REPORT 724
FIGURE 581 TABLE DIFFERENCE REPORT RIGHT CLICK MENU 725
FIGURE 582 SELECTING REVISIONS TO COMPARE 726
FIGURE 583 SUCCESSFUL RUN OF A TABLE COMPARISON REPORT 727
FIGURE 584 PAGE 2 OF TABLE COMPARISON REPORT 728
FIGURE 585 VIEW WHERE USED REPORT 729
FIGURE 586 IN-USE LOCKING ON AN ELEMENT 731
FIGURE 587 IN-USE MESSAGE 732
FIGURE 588 IN-USE LOCK INFORMATION 733
FIGURE 589 LOCKED PROGRAM VERSION BANNER 734
FIGURE 590 SELECTING A PROGRAM VERSION TO LOCK 734
FIGURE 591 LOCKING A PROGRAM VERSION 735
FIGURE 592 SELECTING A PROGRAM VERSION TO UNLOCK 736
FIGURE 593 UNLOCKING A PROGRAM VERSION 736
FIGURE 594 TEMPLATE LOCKS 737
FIGURE 595 NOTES ON A PROGRAM VERSION 738
FIGURE 596 ADDING A NOTE TO THE MAIN NOTES SCREEN 740
FIGURE 597 COMPLETING THE NOTES DETAIL SCREEN 741
FIGURE 598 DIALOGUE BOX 742
FIGURE 599 COMPLETED NOTES SCREEN 743
FIGURE 600 PDF FILE IN THE INSBRIDGE DOCUMENT VIEWER 744
FIGURE 601 EDITING A NOTE 745
FIGURE 602 DELETING A NOTE 746
FIGURE 603 CHANGE CONTROL 748
FIGURE 604 DELETING A CHANGE CONTROL 749
FIGURE 605 CUSTOM VALUE 750
FIGURE 606 SELECTING TRUSTED SITES 753
FIGURE 607 ADDING RATEMANAGER TO TRUSTED SITES 754
FIGURE 608 DELETE FILES 758
FIGURE 609 SELECTING FILES TO DELETE 759

Welcome to the *Oracle Insurance Insbridge Enterprise Rating RateManager User Guide.* This guide describes the features and functionality of Oracle Insurance Insbridge Enterprise Rating RateManager (RateManager). RateManager is a component within the Oracle Insurance Insbridge Enterprise Rating (Insbridge) System that enables users to manage the product definition and modification process, including premium calculation and underwriting rules management.

AUDIENCE

This guide is intended for RateManager users and system administrators who are tasked with creating rating and underwriting logic and managing system features. Users should be familiar with their company's practices and terminology. Advanced users may benefit from having knowledge of their company's current working rate manual.

RELATED DOCUMENTS

For more information, refer to the following Oracle resources:

- The Oracle Insurance Insbridge Enterprise Rating Framework Administrator User Guide.
- You can view this and other Insbridge guides on-line at this address:

  http://www.oracle.com/technetwork/documentation/insurance-097481.html

CONVENTIONS

The following text conventions are used in this document:

Convention	Description
bold	Boldface type indicates graphical user interface elements associated with an action.
Italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`Monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

NAVIGATING RATEMANAGER

Navigate through Oracle Insurance Insbridge Enterprise Rating RateManager (RateManager) using the RateManager top and side menus as well as right click menus, which will be pointed out as they occur. Browser tool bars may not be displayed, may not navigate correctly, or may try to close your RateManager instance.

If you have to visit another web site, open another window. It is a good practice to save your work before you visit another web site or leave your desk.

NOTE: *Insbridge requires Microsoft Internet Explorer 11. Other browsers are not compatible at this time.*

NOTE: *If you have a popup blocker installed, you need to disable it for RateManager to work properly. See Appendix A for enabling popups.*

BASIC SAFETY PRECAUTIONS

It is recommended that you secure your computer when you step away. This prevents any unintentional deletions or entries and protects the integrity of your work.

INACTIVITY/TIMEOUT

RateManager, a web-based application, logs you out without saving your work if you are inactive for a length of time. Your system administrator establishes the specified time. The default setting is 5 hours. If timeout does occur, log back onto the system and continue to work.

SYSTEM REQUIREMENTS

For minimum operating system and hardware requirements, please see the Hardware Software requirements guide.

Guide History

New editions incorporate any updates issued since the previous edition.

Edition	Publication Number	Product Version	Publication Date	Comment
1st Edition	P01-740-15	R 4.0	April 2010	New GUI/navigation
2nd Edition	P01-740-16	R 4.0.1	August 2010	Updates
3rd Edition	P01-740-17	R 4.1	December 2010	Updates
4th Edition	P01-740-18	R 4.5	May 2011	Updates
5th Edition	P01-740-19	R 4.5.1	September 2011	Updates
6th Edition	P01-740-20	R 4.6	May 2012	Updates
7th Edition	P01-740-21	R 4.6.1	November 2012	Updates

8th Edition	P01-740-22	R 4.7	September 2013	Updates
9th Edition	P01-740-23	R 4.8	August 2014	Updates
10th Edition	P01-740-24	R 4.9	December 2014	Updates
11th Edition	P01-740-25	R 5.0	August 2015	Updated Revisions
12th Edition	P01-740-26	R 5.1	December 2015	Updated Revisions
13th Edition	P01-740-27	R 5.2	July 2016	Update Revisions
14th Edition	P01-740-28	R 5.3	October 2016	Update Revisions
15th Edition	P01-740-29	R 5.4	January 2017	Update Revisions
16th Edition	P01-740-30	R 5.5	September 2017	Update Revisions
17th Edition	P01-740-31	R 5.6	Auguest 2018	Update Revisions

RateManager allows users to manage rate programs through the definition of variables, algorithms and underwriting rules. Various classification levels allow you to organize programs and share elements. The RateManager system uses the same overall structure for any rate program. A domain can hold as many projects as needed. Each project holds one or more products. Within each product, you can have an unlimited number of programs under an unlimited number of folders. Within each program, there can be an unlimited number of elements that can be used in algorithms and underwriting rules to produce a specific result.

Prior to working in RateManager, you may find it helpful to review a rating manual that contains the rules and premium logic associated with the rating programs you are going to create. The rate manual should define what is to be included in those base rates. Also, you may also want to review any applicable underwriting manuals.

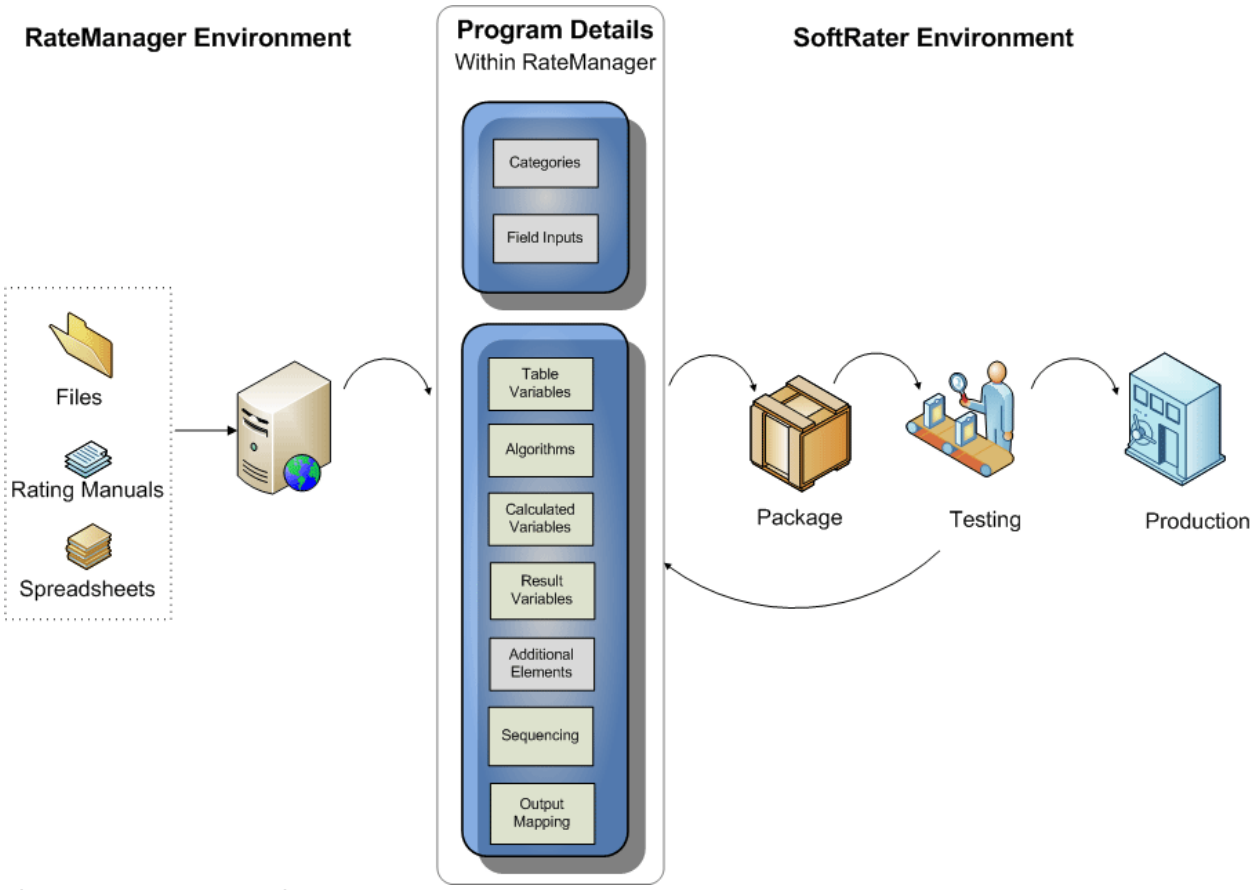


Figure 1 Program Creation Process

Inside RateManager Environment – Files, rate manuals, and other source materials are used to create the categories, fields, and variables that make up a program. When a program is complete, a package is created and is ready for testing.

Inside SoftRater Environment – A package is a snapshot of a program as it is at the time the package is created. Packages can be tested using user created Test Case files and returned to RateManager for further modifications. When a package is ready, it is loaded in to the Production environment for use.

DATA PREPARATION AND PROGRAM STRUCTURE

Data preparation and program structure are crucial to minimizing the amount of time it takes to create a program, as well as determining how easily the program can be maintained. The following points may help you organize what variables need to be created and provide recommendations on setting up tables that can help the program to run at optimal efficiency.

1. Start with the rating manual. Highlight all data elements that need to be loaded into the program. This may include:

   - Steps to be taken
   - Order of the steps
   - Variables and factors
   -
   - Calculations
   - Rate tables
   - Risk Items

2. After you have defined what is to be included in those base rates, you need to determine how you want to organize the shared and common elements and how many unique elements you may need.

3. Another consideration is how you want to define and group your programs. RateManager allows for a detailed, highly separated classification or a simple, flat classification of programs. A flatter classification allows for more shared elements. A separated classification allows for more unique elements to be used.

For Example

A Homeowners Base Rate may include:

- Home/dwelling: This may be a flat dollar amount (In RateManager this would most likely be a Field)
- Other structure: This may be a percentage of home/dwelling (Calculated Variable)
- Contents/Personal Property: This is a percentage of home/dwelling (Calculated Variable)
- Loss of Use: This may be a percentage of home/dwelling (Calculated Variable)
- Liability: This may be a flat dollar amount (Field or Table Variable)
- Medical Payment: This may be a flat dollar amount (Field or Table Variable)

The home dwelling dollar may be calculated on:

- Year built (Field)
- Location/Territory (Field or Table Variable)
- Protection Class (Field or Table Variable)
- Construction (Field or Table Variable)

It is also recommended that you have an electronic version of your data tables. This may help you avoid re-entering data.

1. List all data that can be grouped together in a single table. Examples would be violations and factors based on the same inputs, such as model year factors. Table variables can be linked together to share a single table (linked variables). Linking together variables has the following benefits:

- A single table needs to be built and imported one time into RateManager.
- As changes are made to the program, only one table requires alteration for all the variables linked to it.
- Decreases rating time because the program has to access the database less.

2. Outline all algorithms that are needed to produce rates for the program.

3. Review the algorithms for factors that need to be calculated and are not passed as an input. A class factor is usually determined by several other inputs. Listing out all elements needed to create this variable ahead of time will insure that all calculated variables are completed prior to running the algorithm section.

4. Create your tables. Each table must be imported in either an Excel workbook format a tab-delimited format. The data columns need to be entered so that a result is created. In the Example Excel Table, the last column, **Territory,** determines the result value in the first column, **BI.** For example, if the **Territory** is 1, then the **BI** would be 95.

Example Excel Table

BI	Territory
95	1
103	2
131	3
121	4

5. Determine if the homeowners product share common elements with another product, for example, such as property or flood. Property and flood may need or use the same information such as:

   - Year built (Field)
   - Location/Territory (Field or Table Variable)
   - Construction (Field or Table Variable)

   Sharing information may indicate that the three products can be grouped under a single project. If you need or want, you can create a separate project for each product.

Restrictions

While RateManager is flexible and accommodating, there are a few restrictions.

Name Entries

Most name entries, variables, algorithms, fields, folders and programs, etc., are limited to 40 characters including spaces and most cannot contain special characters. If you enter a special character you may receive an error message.

Restricted Characters are:

Name	Character
Ampersand	&
Apostrophe	`
Asterisk	*
Colon	:
Greater-Than Sign	>
Less-Than Sign	<
Question Mark	?
Quotation Mark	"

Notes and Change Controls:

Notes and Change Control restrict:

Name	Character
Ampersand	&
Asterisk	*
Backslash	\
Greater-Than Sign	>
Less-Than Sign	<
Question Mark	?
Quotation Mark	"

All other characters are allowed.

Program and Program Version Names:

Program and Program Version names restricts:

Name	Character
Pound Sign	#

All other characters are allowed.

Set Message Step:

The Set Message Step restricts:

Name	Character
Ampersand	&
Apostrophe	'
Asterisk	*
Backslash	\
Greater-Than Sign	>
Less-Than Sign	<
Quotation Mark	"
Slash	/
Vertical Bar	|

All other characters are allowed.

Digit and Decimal Separators:

Digit and decimal separators restricts:

Name	Character
Ampersand	&
Apostrophe	'
Dollar Sign	$
Greater-Than Sign	>
Less-Than Sign	<
Pound Sign	#
Quotation Mark	"

All other characters are allowed.

Table Variables:

When working with Table Variables, there are differences in where special characters are allowed and are not allowed.

Table variable data files are the contents inside the file to be imported into a table variable or exported out of RateManager. Table Variable Data Files will be rejected if restricted characters are contained within the file.

Restricted Characters are:

Name	Character
Ampersand	&
Apostrophe	'
Greater-Than Sign	>

Less-Than Sign	<
Plus Sign	+
Pound	#
Quotation Mark	"
Tilde	~
Bar + H	|H
Bar + h	|h
Bar + L	|L
Bar + l	|l

All other characters are allowed in a table variable data file.

Default Table Variable Values:

A table variable default value is the value entered as the default for the table. Table Variable Default Value restricts:

Name	Character
Apostrophe	'

All other characters are allowed in a table variable default value.

Table Variable File Names:

A table variable file name is the name of the file that will be imported or into RateManager or exported out of RateManager. For example, 'BI_Rates4-2011.txt'. Table Variable File Names restricts:

Name	Character
Ampersand	&
Apostrophe	'
Asterisk	*
Backslash	\
Colon	:
Greater-Than Sign	>
Less-Than Sign	<
Question Mark	?
Quotation Mark	"
Slash	/
Vertical Bar	|

All other characters are allowed in a table variable file names.

Updating Variables with Special Characters:

In previous releases of RateManager, some special characters that are now restricted were allowed, for example the asterisk (*). If you attempt to edit an element that uses a special character, you will receive an error. The error message will instruct you to not use the special character.

You can continue to use the element as it is, but you will not be allowed to edit it. If you want to edit the element, you will have to remove the special character.

Elements with special characters that are no longer allowed cannot be updated. Elements that were created with special characters can continue to be used and will not have the special characters removed.

Custom Mapping:

If using the custom mapping tool in RateManager for output mapping, these characters cannot be part of an element or attribute name in target output document:

Name	Character
Asterisk	*
At Sign	@
Backslash	\
Colon	:
Comma	,
Dollar Symbol	$
Empty Space	' '
Equal Sign	=
Exclamation Point	!
Greater-Than Sign	>
Less-Than Sign	<
Minus Sign	-
Parenthesis	( )
Period (dot)	.
Plus Sign	+
Question Mark	?
Quotation Mark	"
Semicolon	;
Slash	/
Square Brackets	[ ]
Vertical Bar	|

Naming Restrictions

Names should be unique to the area where they are being used to reduce confusion over what element is being used. In many places you will not be allowed to enter a duplicate name, with a few exceptions:

- **Products** – Names must be unique. Duplicates are not allowed.
- **Projects** – It is recommended that names be unique.
- **Folders** – It is recommended that names be unique.
- **Version Folders** – It is recommended that names be unique.

  For folders and version folders, names must be unique at the sibling level but can be duplicated if created at either a parent or child level. For example, if you create a folder called ‘Level1’ you will not be able to create another folder at the same level with that name. You can create another folder underneath ‘Level1’ and also call it ‘Level1’.
- **Programs** – Names are unique per folder. The same program name can be used in multiple folders. In Folder A you can have program ‘Jan’. You also can have program ‘Jan’ in Folder B.
- **Version Names** – Unique per program. The same name can be used in multiple programs. In Program1 you can have version ‘Alpha’. You also can have version ‘Alpha’ in Program 2.
- **Fields** – Names are unique names per project.
- **Result Variables** – Names are unique per project.
- **Table Variables** – Universal table variables are unique to the project. Global table variable names are unique per product. Local table variable names are unique per program.
- **Calculated Variables** – Universal calculated variables are unique to the project. Global calculated variable names are unique per product. Local calculated variables are unique per program.
- **Algorithms** – Universal algorithms are unique to the project. Global algorithm names are unique per product. Local algorithm names are unique per program.
- **Driver Assignment** – Global driver assignments names are unique per product. Local driver assignment names are unique per program.
- **Constants** – Universal constants are unique to the project. Global constants names are unique per product.
- **Filter Rules** – Universal filter rules are unique to the project. Global filter rules names are unique per product. Local filter rules are unique per program.
- **Validations** – Names are unique names per project.
- For table variables, calculated variables, algorithms and driver assignments, global names and local names cannot conflict. For example, if you have a global element called ‘Rate’, you will not be able to create a local element called ‘Rate.

- Element names are not verified with other variables and algorithms. For example, if you have a field called ‘Manager’ you will be allowed to create an algorithm called ‘Manager’ as well. Each element has an identifying icon to indicate what kind of element it is.

- **Releases, Test Cases, Impact Analysis Reports, etc.** – Names are all unique per module. You can use the same name across modules.

RateManager naming conventions along with distinct icons that can visually assist you in identifying what kind of variable or algorithm is being displayed and whether an element is local or global provide a great amount of flexibility and will allow for most companies to utilize their standard naming conventions.

UNICODE

Unicode allows for multiple language character sets to be used in RateManager. The screens will still be in US English and field values will not be translated. Unicode means that the editable fields on the creation, editing, copying and deleting screens will now be open to the language character set of your choosing. You will be able to enter values in a multitude of languages using a variety of keyboards as long as the keyboard functions in an Internet Explorer environment and the language is recognized by Unicode.

Character Validation

Character validation will be performed to make sure that special characters are not being used. Special characters are the characters that are commonly used as punctuation markers or shorthand symbols such as ampersand (&), exclamation point (!) and caret (^) or commonly used currency and math symbols such as dollar sign ($), equals sign (=) and pound sign (#). Special characters will be restricted as in previous releases with exceptions for use in certain areas.

For example, the at sign (@) will be allowed in the email field when creating users. Special characters are the common US English special characters. Special punctuation unique to certain language sets, for example, the inverted question mark (¿) will not be restricted. However, these special characters may result in errors and should not be used.

Unicode does not require any additional settings or actions at installation.

Insbridge supports UTF-8, which is a multibit character encoding that allows for every character in the Unicode character set to be represented.

Calculations

Calculations using integer, decimal and date values must use the numerals 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. This is only for calculations. Any other character will be recognized as a string and calculations will not be performed. For example, if you attempted to calculate a value using Roman numerals, you would receive an error message stating that a mathematical instruction cannot be performed on a non-number item. You can use Roman numerals in any field that accepts a string value.

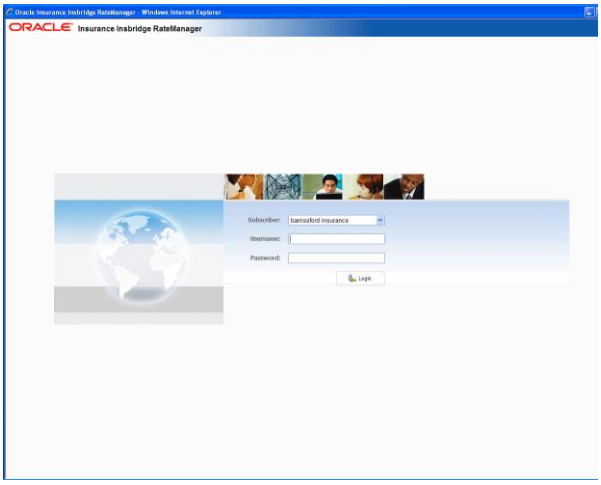
LOGGING INTO THE SYSTEM

Prior to accessing RateManager, you should have received:

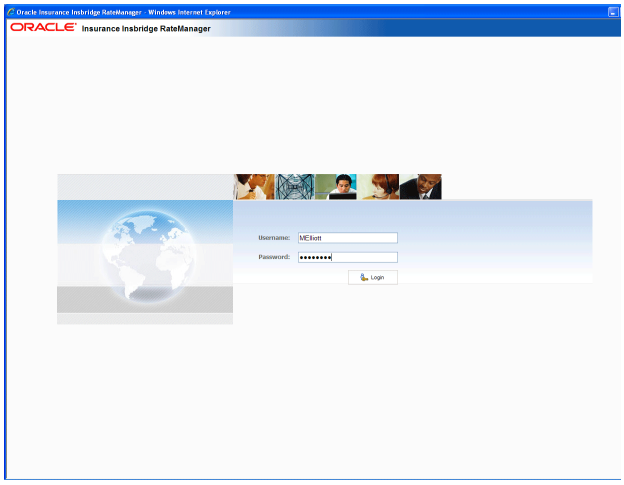
- **RateManager URL:** RateManager can be accessed by going to **http://<yourserver>/RM**, where <yourserver> is the name of the server where RateManager is installed.
- **Username:** usernames are required.
- **Password:** passwords are required.
- **Subscriber:** subscriber information may not be required.

A subscriber may be a single company, a group within the company, or a single person. If more than one subscriber has been set up by the system administrator, you will have to select the subscriber where you want to work. If only a one subscriber has been setup, you will not have to select a subscriber.

If you do not know any of this information, please contact your system administrator.



If multiple subscribers have been setup, you must select the subscriber where you want to work.



A single subscriber has username and password fields only. Subscriber information is not required.

Figure 2 Login Screen

Logging In

Access to RateManager is gained through a login screen. To *login*, enter your username and password and click **LOGIN**. Depending on your company setup, you may need to select your subscriber. If a company subscriber is required, a menu is displayed. Select the subscriber you need, and continue to log in.

A successful login places you on your Home Tab.

A failed login shows an error message. Please correct any errors and try again. If you cannot log in, please contact your system administrator.

Help Logging In

If you have forgotten your password, username or need assistance with logging into the system, please contact your system administrator.

Rights

RateManager uses rights to allow users access to various areas. The type of rights you have determines what you can view, edit, add or delete. There are three types of rights:

- **No Access** – you cannot enter this area. The options or entries are grayed out or not displayed.
- **Read Only** – you can view the information but you cannot make any changes. Save, edit or other features may be listed on the toolbars or right click menus, but you will not be able to effect any changes.
- **Full Access** – you can view, edit, add or delete information where applicable.

The system administrator sets group rights. If there is an area you cannot enter or edit, and you think you need access, please contact your System Administrator.

Logging Out

The Logout option logs you out and closes RateManager.

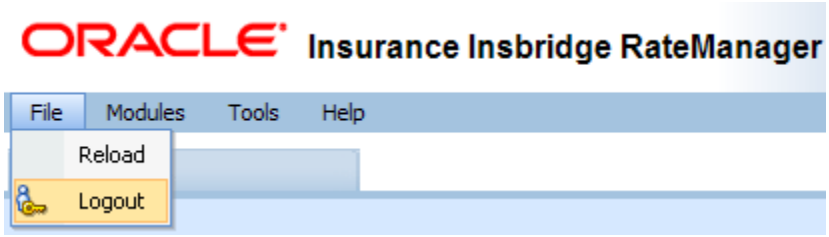


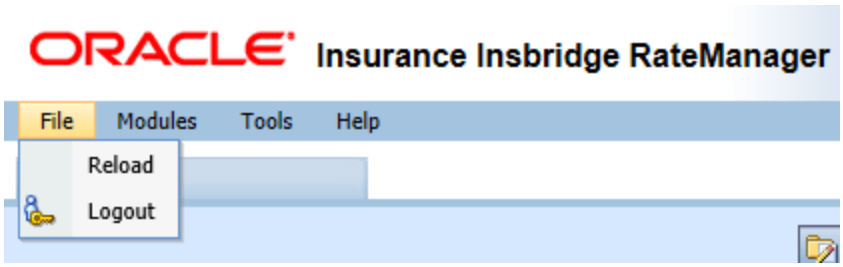
Figure 3 Logging Out

To log out, click File→Logout. If you have any elements open that show a change, you will be asked if you want to save. After saving your work, you will be logged out.

RATEMANAGER TABS AND MENUS

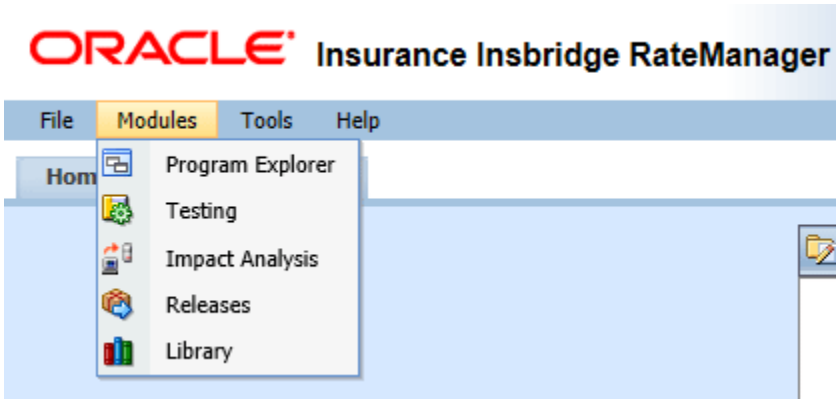
RateManager allows for multiple tabs to be utlized. The tabbed authoring interface allows for multiple activities to be open across separate tabs without losing any functionality. Each tab functions independently. You can view the Program Explorer, then edit a program version, and then create a new algorithm without being forced to close the prior tab.

The RateManager tabs follows a familiar model, a fixed top bar menu with four options. Each option contains a menu of system features. Each feature opens either a new tab or a window.



The File menu:

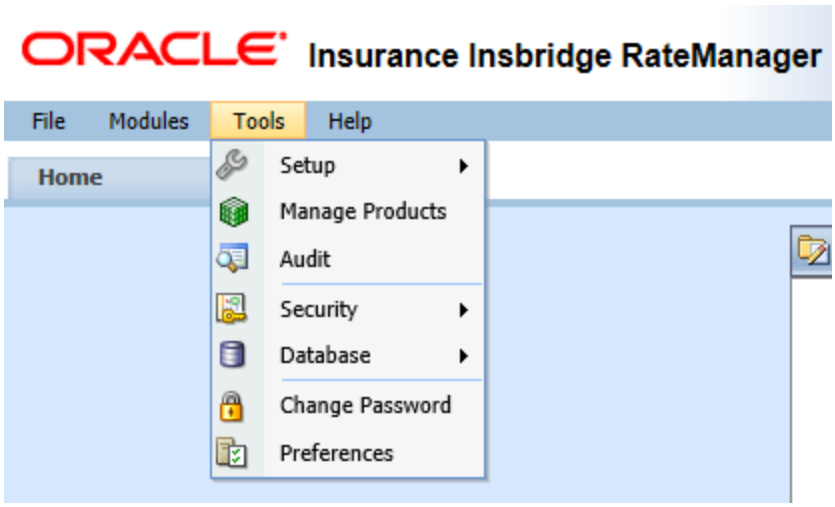
- **Reload** – reloads the page. Use this option when you want to refresh the screen.
- **Logout** – logs you out of the system.



The Module menu:

- **Program Explorer** – the program explorer serves as a central location for creating and managing programs and accessing program elements.
- **Testing** – the testing module is where you can test your rating program during development without loading packages.
- **Impact Analysis** – the area for batching and data analysis.
- **Releases** – allows groups of rating packages to be automatically deployed and loaded to IBFA through a release. If you do not have release rights, this option is not displayed. Please see the RateManager Modules User Guide Volume 2 for more information.
- **Library** – the library is a central repository within RateManager where export programs, template programs, public XML files, custom XML, and table data export files are stored and managed.

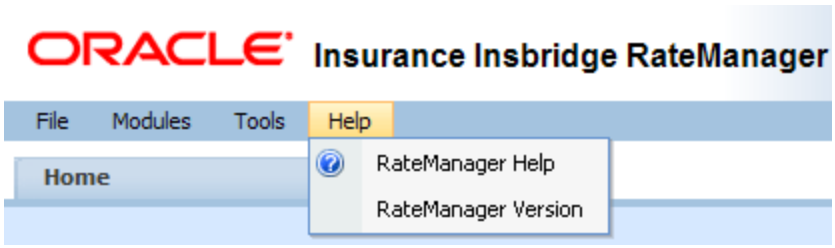
This guide focuses on the features and functionality in the Module menu.



The Tools menu:

- **Setup** – expands for access to the Naming Standards option, System Level Constants, Message Templates, Reporting Data Model information, and Table Job Management.
- **Manage Products** – administrators can manage products. This is an advanced system feature and may not be available to all users.
- **Audit** – contains the audit logs of system users. This is an advanced system feature and may not be available to all users.
- **Security** – is where users, user groups and sessions are managed. This is an advanced system feature and may not be available to all users.
- **Database** – this area is for administrators to create backups and deploy updates. This is an advanced system feature and may not be available to all users.
- **Change Password** – allows users to change their password.
- **Preferences** – brings up the window where you can select the snaps you want to display on your Home Tab. For advanced system users, system setup information is displayed.

For help with Tools menu, please see the RateManager User Guide: Tools.

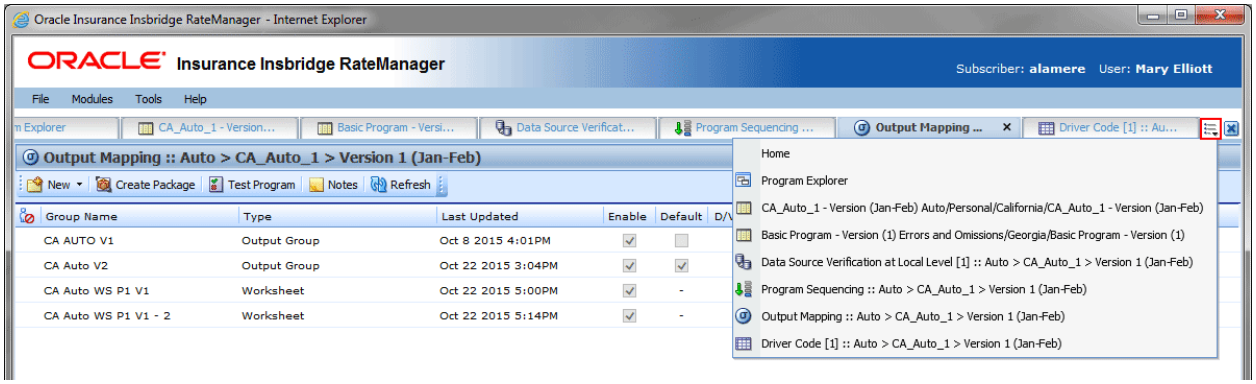


The Help Menu

- **RateManager Help** – clicking this option brings up online help.
- **RateManager Version** – clicking this option brings up system information.

Tab Navigation

Multiple tabs can be open at any time. You can move from tab to tab by clicking the tab you want. You also can click the tab display icon in the upper right of the screen to see a list of all the open tabs. You can navigate to any tab in the list by selecting it.



NOTE: *If you have a window open, you have to close it before you can click another tab.*

- The close tab icon closes the tab you are currently on. If you have any elements open that show a change, you will be asked if you want to save.

Keyboard Shortcuts

In a few areas, you cannot use your mouse right click features to cut and paste however, you can use your keyboard controls. Highlight the text you wish to cut or copy. Use the Control Key plus the letter key to either “C” copy or “X” cut. Position your cursor where you want your text to be placed. Use the Control Key plus the letter key “V” to paste.

- **Control + X** cuts text.
- **Control + C** copies text.
- **Control + V** pastes text.

- To go to the beginning of an entry, use the home key on your keyboard.
- To go to the end of an entry, use the end key on your keyboard.

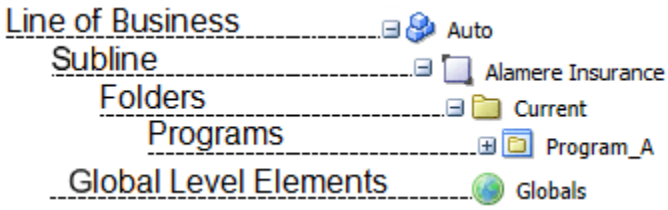
NOTICE FOR UPGRADE USERS

If you are upgrading from a previous release to release 5.0 or later, there's a few changes to be aware of.

The Home tab and some of the modules features, Impact Analysis, Testing and Releases, do not have significant changes. Functionality and appearance for these tabs remains the same. Where you will see significant changes is in Program Explorer. All the features and functionality you expect are still here but the look and navigation have been updated.

Previous Releases

Previous releases program structure consisted of:



This structure reflected active lines of business with a subscriber named default subline underneath where folders were placed to hold programs. Global elements were available to every program in the subline.

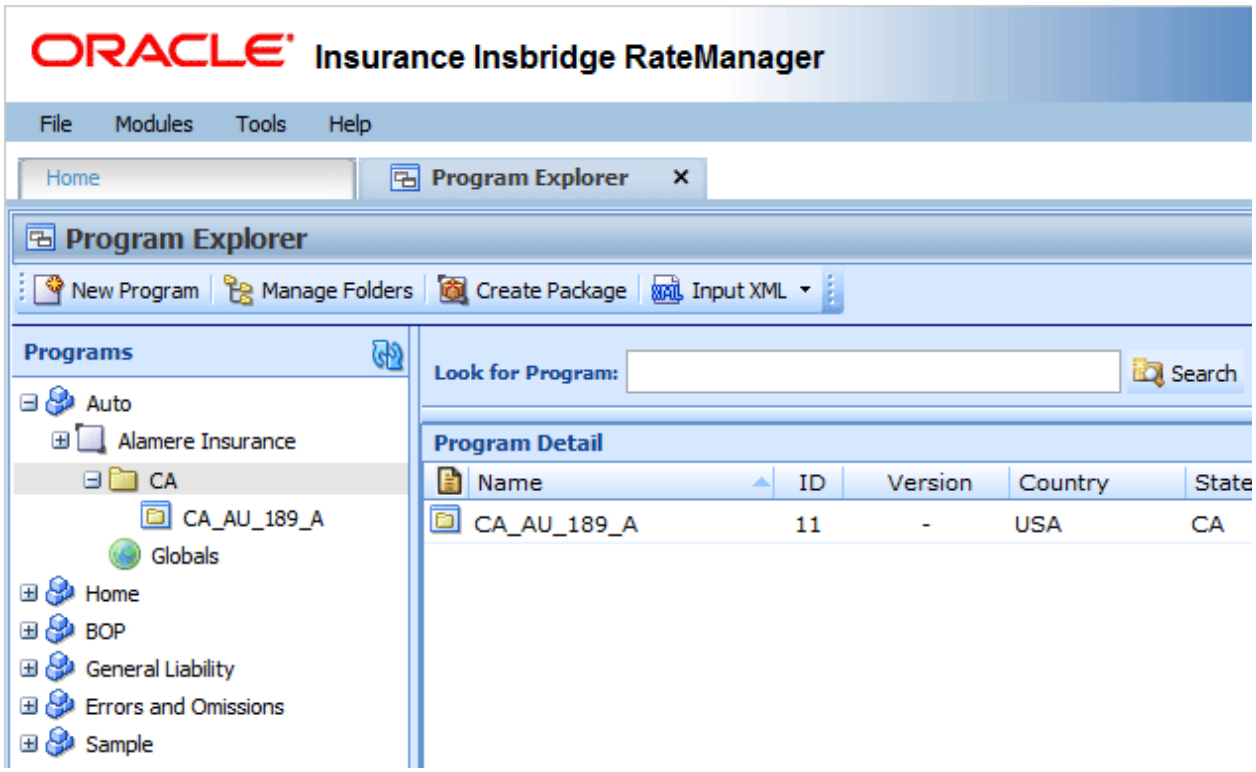


Figure 4 Program Structure in Release 4.9

Release 5.0 and After

After upgrading, the same elements are present on the Program Explorer tab along with a new structure and new elements.

- A new element is the domain type, which is RateManager. This indicates the source of the content. All updated release content is located in the RateManager domain.
- A new element is the domain. For upgrade users the domain, each active line of business is created as a separate domain in the 5.0 release. This reflects the previous structure where each line of business had a unique category and element structure with no shared elements.
- A new element is the project. Under each domain, a project has been created using the current subline name. The default subline name in a project cannot be edited. This reflects the previous structure where subscriber named sublines where automatically created under lines of business. Also new at this level is universals. Universal elements can be used by any program in the project. Categories and fields, now called fields, have been located here.
- The product is also a new element. Under each project is product that reflects the line of business used in previous releases. Global elements are still available to every program but are now found at the product level.

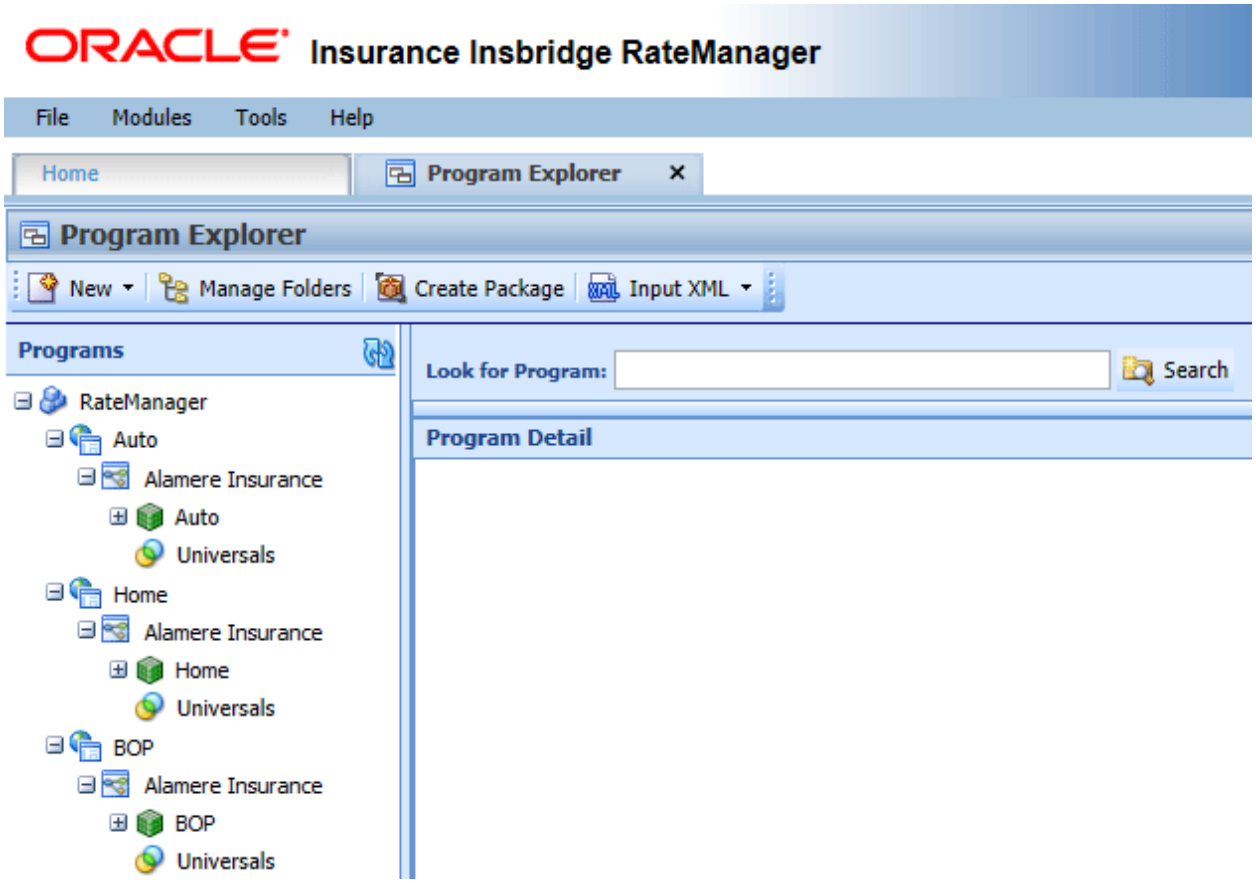


Figure 5 Program Structure in Release 5.0

Folders are the same. Folders are located under products and programs are still found in folders.

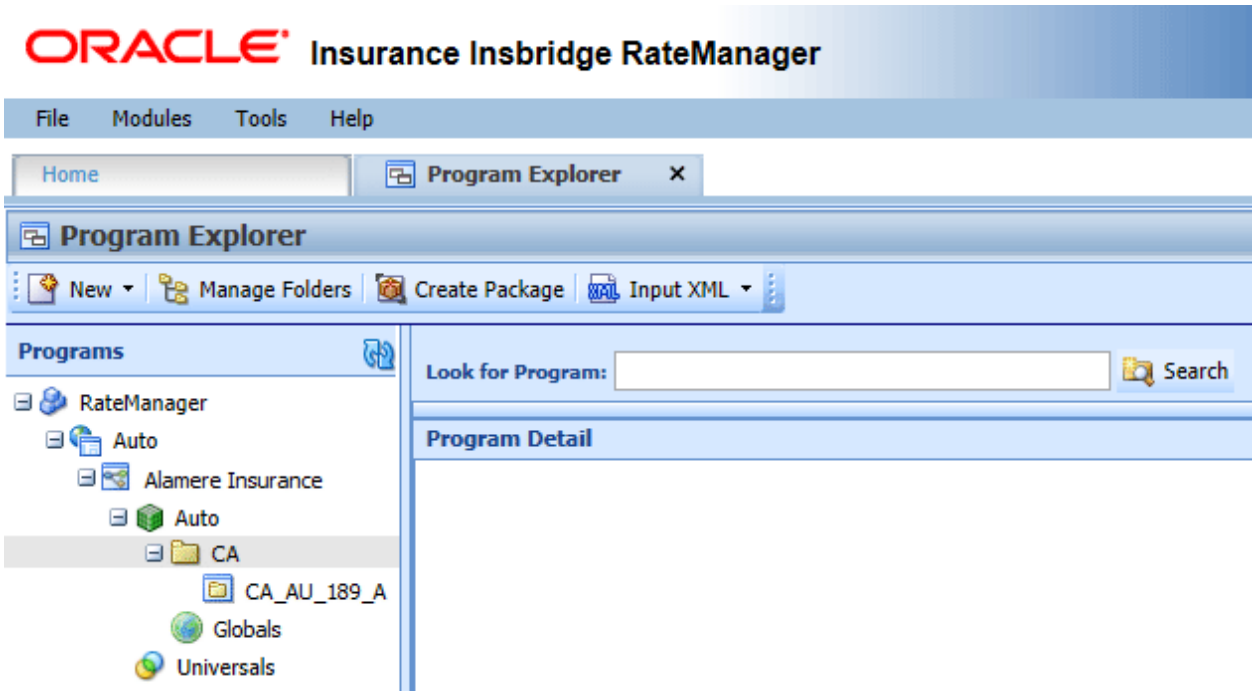


Figure 6 Expanded Program Structure in 5.0

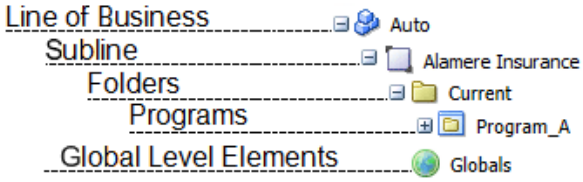
Program creation is done on the program details tab. Creating and managing program elements is much the same with the addition of a new element, filter rules.

Program Structure Before and After

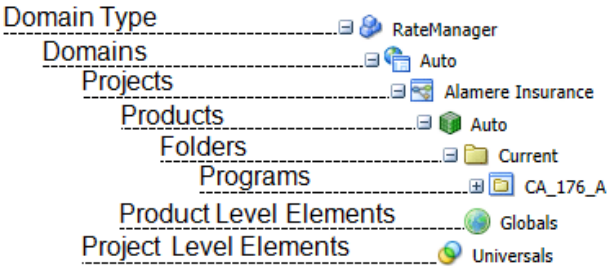
All program elements from the upgrade release have been brought in to the upgraded release.

The expanded capabilities and wide ranging features still give insurers the strength, flexibility, and capacity that they need but Insbridge Enterprise Rating also can be utilized by other types of insurers who require support for multiple lines of business, multiple product offerings, multiple rating formulas, and the ability to navigate complex business rules quickly and efficiently.

RateManager Program Structure for Release 4.9 and earlier



RateManager Program Structure for Release 5.0 and later



RATEMANAGER CONSTRUCTS

RateManager has a few constructs that you may want to take into consideration when creating your program.

Order of Operations

RateManager does not use the standard arithmetic order of operations to perform calculations on elements that are not grouped. Instead, RateManager performs calculations left-to-right or top-to-bottom.

For example, here is how the standard equation 5 + 2 x 3 would be evaluated using the standard order of operations, versus how it would be evaluated in RateManager:

Standard Order of Operations: 5 + (2 x 3) = 5 + 6 = **11**

RateManager: (5 + 2) x 3 = 7 x 3 = **21**

The standard order of operations is for multiplication to be performed first and then addition. In RateManager, the first equation to be performed is addition and then the multiplication.

If parentheses are being used, for example in an IF Statement step type, the elements in the parentheses will be processed first. After the parentheses have been processed, the execution order returns to the left (or top) and the next element to be processed will be the first equation and so on.

LOCALE

Locale is defined as the unique attributes of a location including the formatting of dates and numbers. The default locale is the locale of the sever where the Insbridge system is installed. Every activated product assumes the default locale. The date mask, the decimal symbol and digit grouping symbol are automatically filled with the defaults from the server. The product (line of business) can select a different locale from the server. The values set at the product level apply as the default locale to every program created in the product however, the program can select a different locale if necessary. The updated locale applies only to the program where the locale was changed.

The locale hierarchy is: the Insbridge server determines the product locale. The product determines the program locale. The locale used by the program is what will be used when rating.

There are two places where you can select the Locale:

- **In the Manage Products screen:** For more on setting a locale in the Manage Products screen, please see the RateManager Tools guide.
- **In Programs**

To verify a locale for the product, enter the Edit Product screen in the Manage Products option. The locale, date and number format are listed for the selected product. The Manage Products option requires admin permissions. Please contact your system administrator if you require access or need information.

To verify the locale for the program, enter Program Properties. The locale the program is using is listed.

Selecting a Locale for the Program

You can select a locale for your program when you create a new program or edit an existing program. These are the same Locale options that are available at the product level. If you do not select a locale, the locale that the product has is used.

The New Program screen and the Edit Program screen contain a drop down of available Locales. The default is the locale of the product. These values can be changed and you are allowed to type in an alternate value if you choose. You must select a value for the Date Mask, Decimal Symbol and Digit Grouping Symbol. A Locale is always listed.

Locale when Packaging

When you package a program (SRP), the Locale that is currently selected is the Locale that the SRP uses. An SRP uses the Locale that has been selected at the program level. If you update the Locale and re-package, the Locale is updated as well. If you deploy the updated package to IBFA or IBSS, and the program version has not been changed, the re-packaged version replaces the previous version.

Locale in Global Variables

When editing a global variable, the Locale settings at the product level are used for validation. When editing a local variable that uses a global variable, the Local settings at the program level are used for validation.

Locale in Table Variables

Table variables use the Locale that is defined in the program. Data entered into a table variable should match the locale selected. For example, if you have created an Effective Date Table and entered all the data in MM/DD/YYYY format, then you should make sure your Locale matches that date format.

If you attempt to input data that does not match the data and number formatting of the program, you may incur errors.

Locale in Templates

In a template, local elements retain the date and number formatting of the system where they were created. Date and number formats for global elements assume the date and number format of the product where they were brought in. This may not be the same as the settings of the system where they were created. Be aware that any global elements may be changed if the product date and number format is different from the date and number format of the originating system.

Locale in Callouts

When using a Program to Program (P2P) callout, date and number formatting must match or the rating will fail.

Changing a Locale

Changing a Locale at the product level affects the default locale on new programs only. Existing programs are not affected by any changes. Existing programs already have a locale set.

Changing a Locale at the program level updates every element used in the program to the new Locale. It is not possible to mix Locales within a program. The selected Locale is currently the Locale that is used when the program is packaged.

Locked programs can have the Locale updated. Locales are copied to the newly created program or program version.

Although Locale can be updated at any time, it is not recommended. Updates to Locale may impact rating and may produce unexpected results.

WARNING: *Changing date and number formats may result in programs not running properly. Global variables that have been created as one type of date and number format may not function as expected with another date and number format. Date and number formats should be consistent through the program.*

Date Masks

The date mask (date format) specifies how SoftRater interprets dates being passed into an input file. The value entered for the mask will be the format used to validate the data that you enter for variables.

Date formats are displayed as entered and will be executed and evaluated as you expect during rating. Date formats are carried through to any new versions or revisions.

Dates must contain:

- 2 digit Month entry MM
- 2 digit Day entry DD
- 4 digit Year entry YYYY or a 2 digit Year entry YY
- One of 3 separators minus sign (-), period (.) or slash (/) Special characters are not allowed. The list of special characters can be found in

- Restrictions.

Date formats are defined in the product. Any program in the product uses the defined format. A different format can be entered in the program.

- Date formats can be in any combination of the mandatory fields, for example MM/DD/YYYY.
- Date format is case sensitive. You must use capital M, D, and Y.
- A separator must be used, i.e. slash(/) MM/DD/YYYY, minus sign (-) DD-MM-YYYY, or period (.) MM.DD.YYYY.

Dates in Upgraded Programs

Previous releases of RateManager allowed for dates to be entered without separators (or delimiters).
If you have upgraded to Insbridge from a release prior to 4.5 and you are using a date mask that does not contain separators you will need to do the following:

1. Change all Program Date Masks to include separators (or delimiters). Change the mask to match the format of the dates entered in the Program logic (tables, version selection, etc.).
2. Adjust any variable that is masking a field variable of date data type. You will need to make sure that the masking will work with the new Date Mask (format) set in step 1.
3. Any dates in an Insbridge Request file will have to match the new Date Mask (format) set in step1.

These steps will need to be done on any program where the date mask does not have separators.

Number Formatting

Number formatting refers to the decimal and digit separators found in common number formats, for example, 1,000.00 or 10.000,00.

- **Decimal Symbol** – The decimal symbol indicates the integral part of a number from the fractional part. For example, 10.25 (using a period as the decimal symbol) or 8,50 (using a comma as the decimal symbol).
- **Digit Grouping Symbol** – The digit grouping symbol indicates the symbol that will be used to show the thousands separator. For example, 1,000 (using a comma as the digit grouping symbol) or 1.000 (using a period as the digit grouping symbol).

  The digit grouping symbol is used for validation purposes and not for display purposes.

Number formats are defined in the product. Any program in the product uses the defined format. A different format can be entered in the program.

- Most separators, i.e. comma (,), period (.), or space ( ) are allowed. Certain special characters are not allowed. A complete list of special characters can be found in Restrictions.

Restricted Special Characters

Name	Character
Ampersand	&
Apostrophe	'

Dollar Sign	$
Greater-Than Sign	>
Less-Than Sign	<
Pound Sign	#
Quotation Mark	"

- You cannot use the same separator for the decimal and digital. The separators must be different.
- Only one separator per entry. You cannot enter a space ( ) for the millions separator and then a comma (,) for the thousands separator. You have to select either the space or the comma.

Default Values

Fields have default values based on the data types that are used. In the event a valid value is not passed through, the default value for the expected data type will be used in rating.

- String is null
- Date is 1501-01-01
- Decimal is 0
- Integer is 0

For example, when testing a program in Test Case Editor, if a value for an integer data type field is not sent in the input XML, then the default value for an integer data type is used. Or if a string type is sent through for a field that is expecting a decimal, the default value for a decimal data type is used.

Parent Child Relationships

RateManager folders, programs, and categories use a parent child hierarchical data model. This is where the top level of data is designated as the *parent* and any level that falls beneath is the *child*. The top most level is designated as the *root*. Children on the same level are *siblings*. A parent child hierarchical data model allows multiple parents to have multiple children all from one root.

The parent child hierarchical data model creates a tree like structure is similar to Microsoft Windows Explorer. This multilevel setup is where you can place an unlimited number of products, folders and programs underneath a project.

VERSIONING

RateManager supports two levels of versioning to allow you to implement rate changes quickly and easily. These levels are:

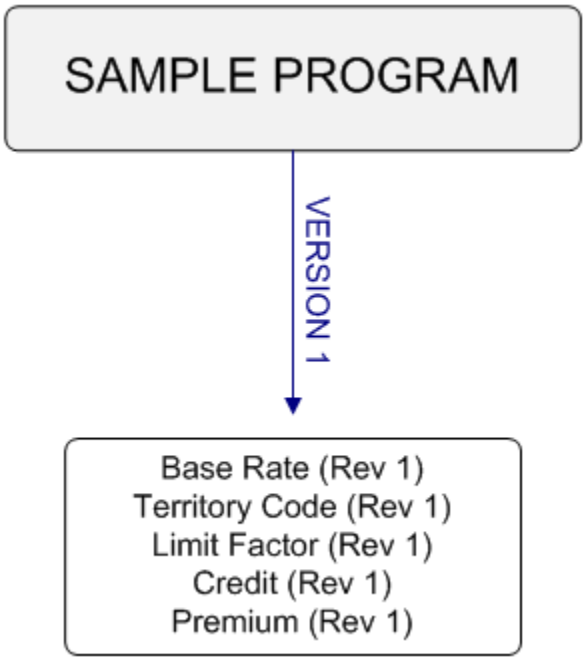
- **Program Versions**
- **Element Revisions**

Program versions and element revisions can be performed on both standard programs and templates programs and are closely related.

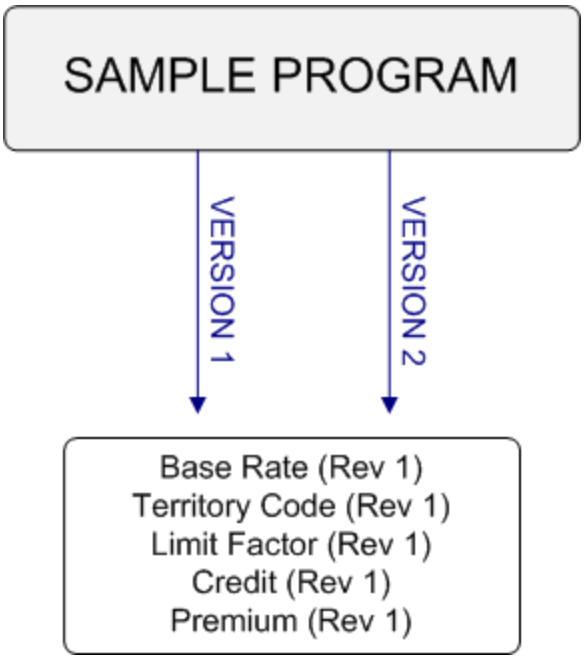
Program Versioning

When a new version of a program is created, it is not a copy of the version it was based on. Instead, the new version is a pointer to the original version and all items that exist in that version.

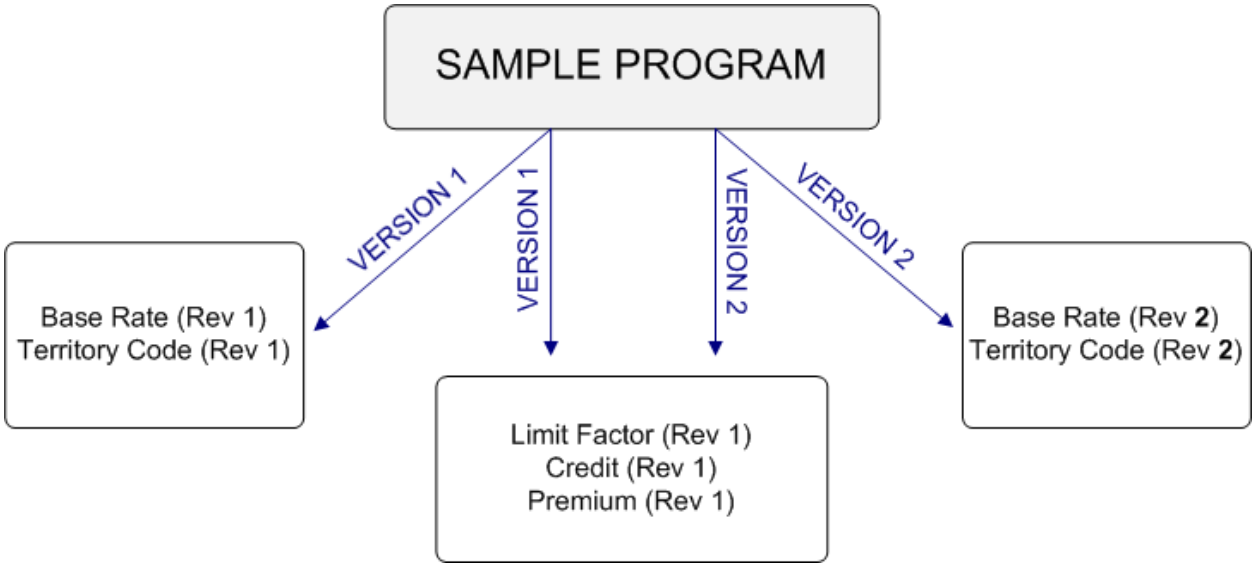
Sample Program represents an initial program before any new versions have been created.



When a second version is created, it initially points to all the same variables and algorithms as the first version.



In order to load a rate change, you will need to create a second revision of all the variables and algorithms that reflect the rate change. In this example, if the Base Rate variable and the Territory Code variable change, you would make a second revision for both of variables. Version 1 would still use revision 1 of all variables. Version 2 would use revision 2 of the Base Rate variable and Territory Code variable, but would still use revision 1 of all the other variables and algorithms.



Element Revisions

All elements of a program can be versioned individually. Figure 7 is an example of revisions for a table variable, shown by using the **Show/Hide Revisions** option on the right click menu of the variable.

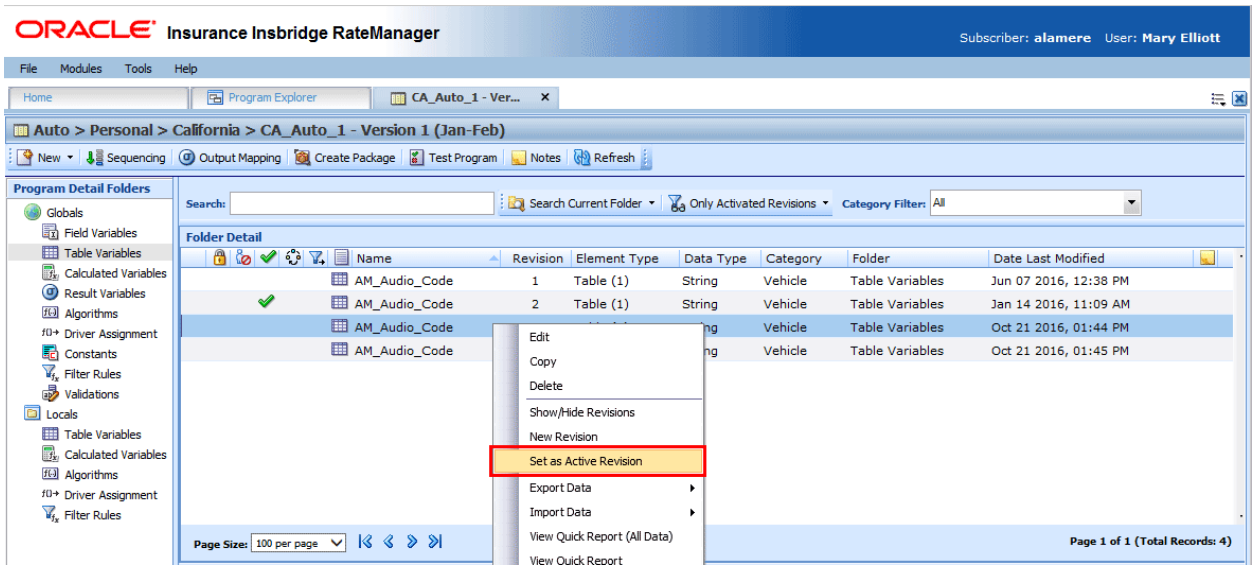


Figure 7 Program Element Revisions

If you wanted the variable to use Revision 3, you would select Revision 3, right click, and select Set as Active Revision. The green checkmark indicates which revision is the active revision. You also can filter results so that only the active revisions are displayed.

Validations have revisions created on the Universal tab. Setting the active revision is done on the Globals tab.

NOTE: *All table variables, calculated variables, algorithms, drivers assignments, filter rules, and validations can be versioned.*

HOME TAB

Once you have logged in, you are on the Home Tab. The Home tab has a control panel that contains links to all the major components of the Module menu and the Setup options from the Tools menu and a help link. Additional panels called *Snaps* also can be added. Snaps are modules that contain information or links to elements that you recently worked on. For example, if you selected the snap **My Recently Opened Programs**, you would see a list of the last 10 programs that you opened.

The Home tab can be customized to contain the information you want to see immediately when you log in. The Home tab is always be the first tab and cannot be closed.

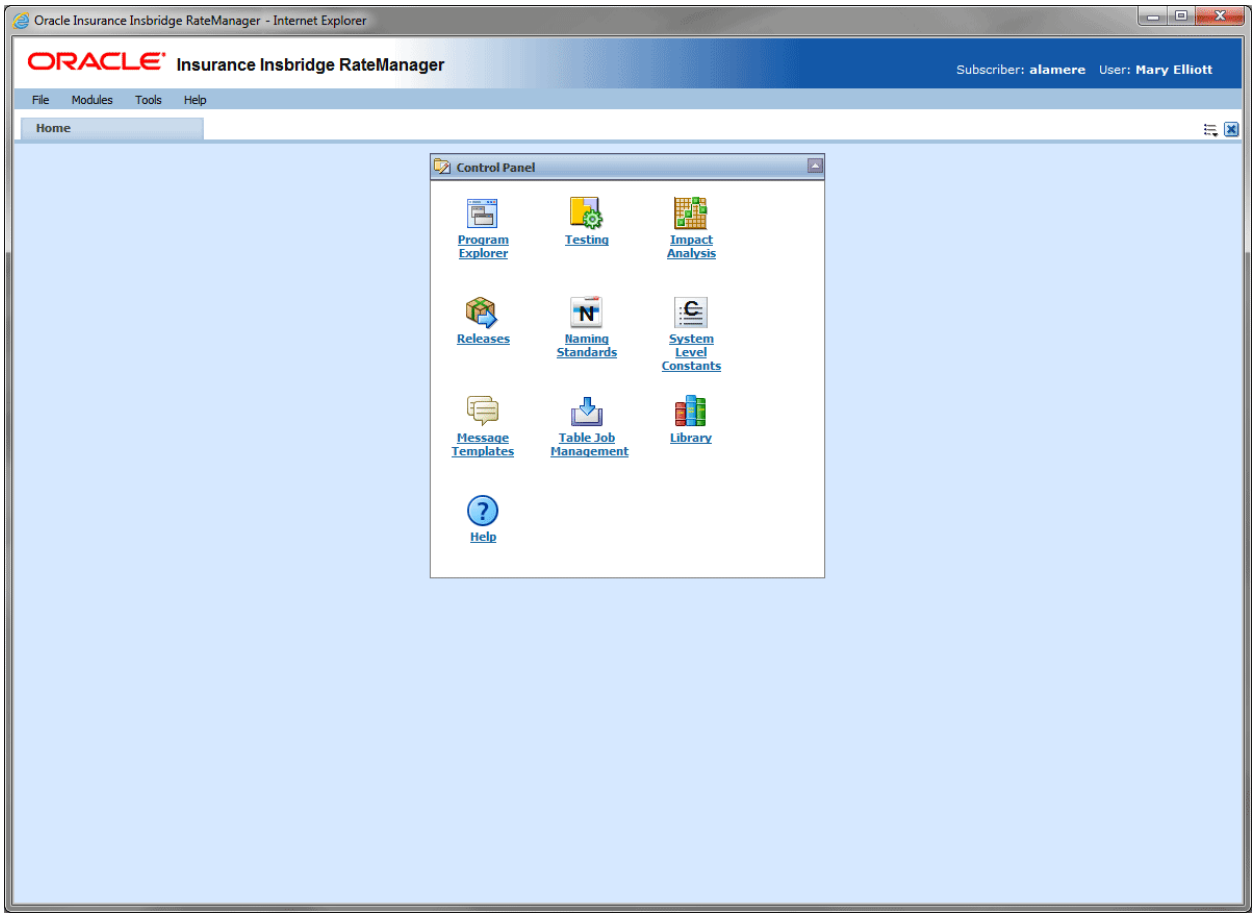


Figure 8 Home Tab

The Control Panel contains links to all the major components of RateManager. This area cannot be removed but can be reduced by clicking the up arrow in the upper right corner of the panel. Depending upon your rights, you can have up to ten links:

- **Program Explorer**
- **Impact Analysis**
- **Testing**
- **Library**
- **Releases**
- **Naming Standards**

- **System Level Constants**
- **Message Templates**
- **Table Job Management**
- **Help**

If you do not have rights to an area, a link is not listed in the Control Panel.

Using Snaps

Inside the snap is a link to the program or element where you recently worked. Double click the name of the program or element and a new tab is opened for the selected item. You also can right click on the link in the snap to open.

You can remove links from the snap by right clicking on the link and selecting to remove from the list.

Snap Icons

There are two icons in the snap top banner:

	If you no longer want the snap, click the close icon. The snap is unchecked from your preferences and no longer displayed on your Home Tab. To place the snap back on the Home Tab, you must enter Preferences and select it.
 	If you want to minimize the snap, click the up arrow. This leaves the snap on the Home Tab but reduces the details. The snap top banner is grayed out. To open the snap back up, click the down arrow.

Positioning Snaps on your Home Tab

Snaps can be repositioned on your Home Tab.

1. Position your cursor over the snap top banner until you get a four arrow cursor.
2. Hold your left mouse button down to select. The snap is ready to move when the entire snap is grayed out.
3. Drag the snap to the location you want. A gray shadow is positioned where the snap will be placed. If the shadow is not visible, you do not have the snap in a location where it can be placed.
4. If this is the location you want, release the mouse button. This drops the snap in place.

Sorting Snap Information

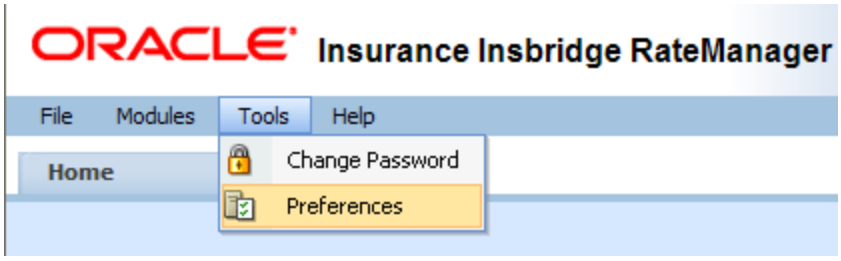
Snap information is listed by Date Last Opened, newest to oldest. To rearrange the information in the snap, click the column header. Information is sorted either alphabetically or numerically, depending on the column header.

Adding Snaps to the Home Tab

Preferences is where you select the snaps you want to place on your Home tab. The Control Panel snap does not have an option to add or remove.

To Select Your Snaps

1. From the top bar menu, select **Tools** and then click **Preferences**.



2. This opens the **Preferences** window.

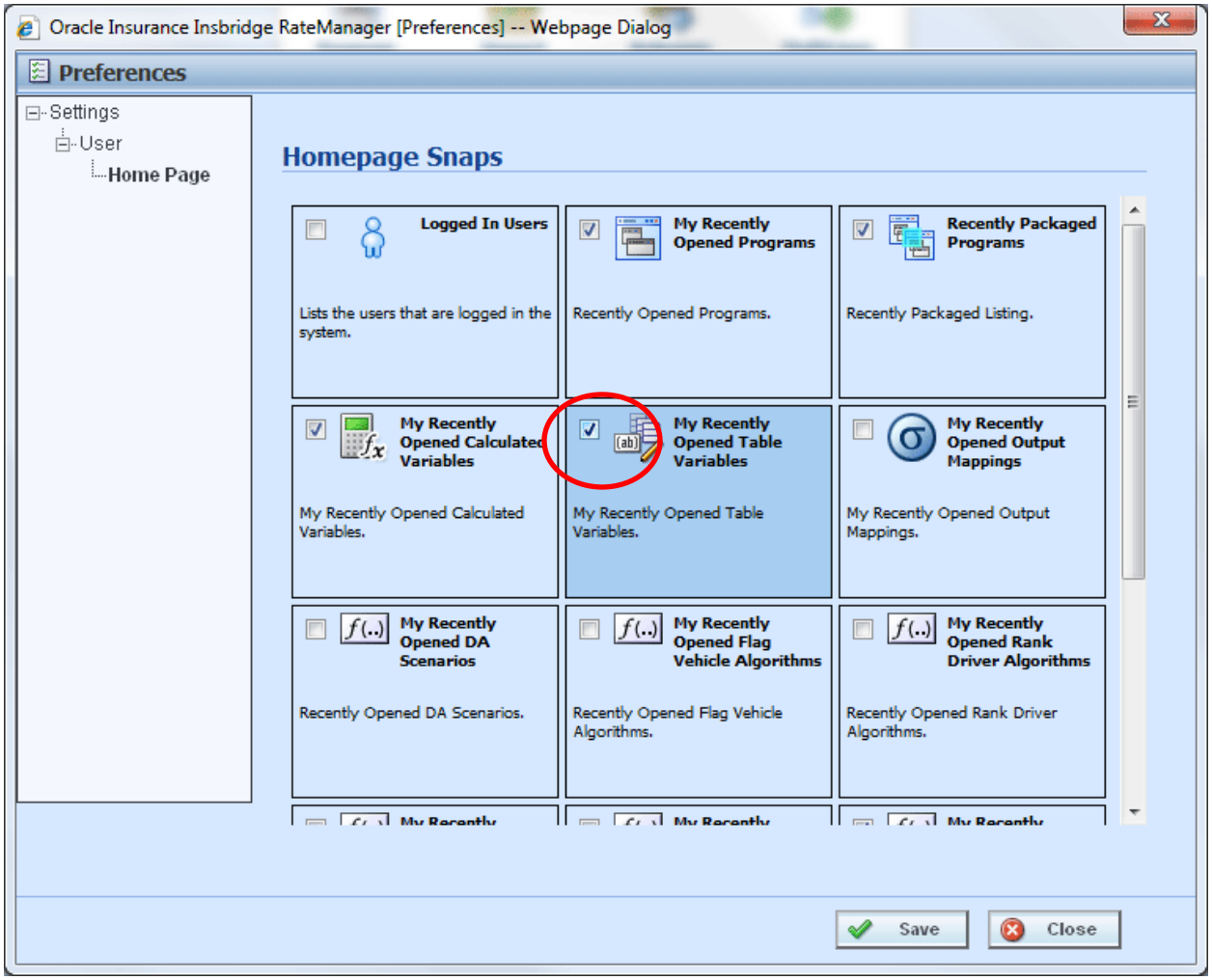


Figure 9 Preferences Screen

3. **Select the Snaps** you want displayed on your Home Tab by checking the box in the corner.

   There are 15 snaps available. 12 snaps contain links to recently worked on elements. These snaps are labeled "*My Recently…*" Opened Programs, Packaged Programs, or Opened Table Variables.

   The "*Logged In Users*" snap lists the users who are logged in to the system right now. The

"*Oracle RSS Feed*" displays the Oracle news feed. This snap may require a proxy setting to be entered by the system administrator in order for it to function properly. If an outside internet connection is not allowed, you will receive an error message. There is also a "*Tutorial Videos*" snap that contains video segments about the features and functionality contained in RateManager.

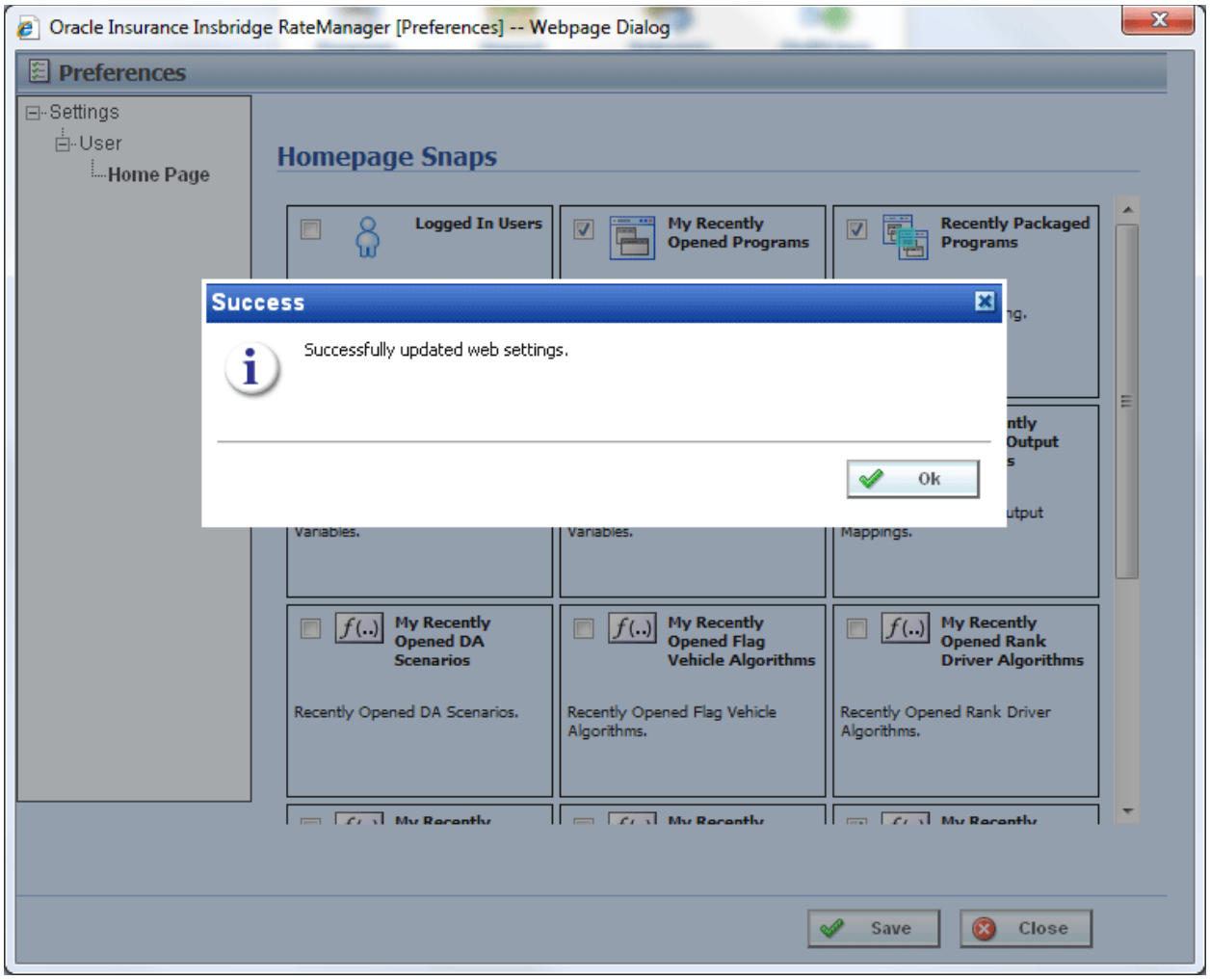


Figure 10 Updated Preferences

4. Click **Save** to save your selections. A success message is displayed.

5. Click **OK** to return to the preferences window.

6. Click **Close** to close the preferences window. Your Home tab now displays the snaps you have selected.

NOTE: *If your Home tab is not displaying your snaps, click File → Reload to refresh the screen.*

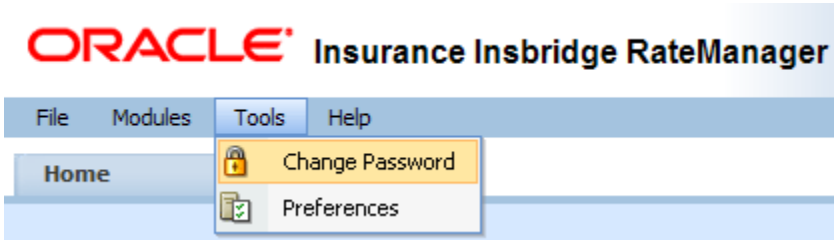
Changing Your Password

You can change your password at any time. RateManager does not prompt users to change their passwords.

It is recommended that you change your password periodically to prevent other users from logging in and making changes using your user ID.

To Change Your Password

1. From the top bar menu, select **Tools** and then click **Change Password**.



2. This opens the **Change Password** window.

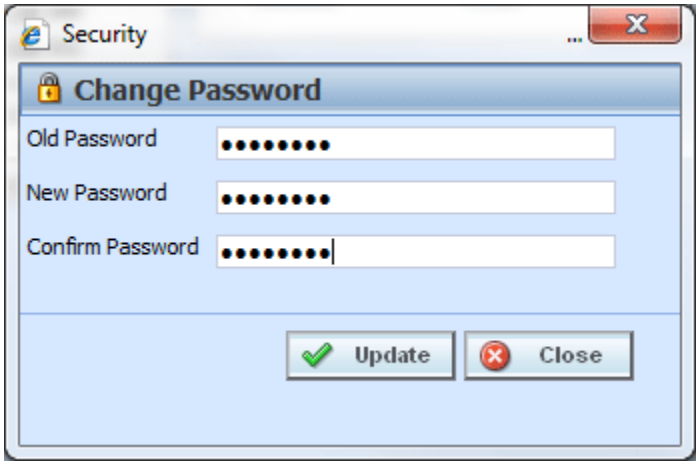


Figure 11 Change Password Screen

3. Enter your current password in the **Old Password** text box and enter your new password in the **New Password** and **Confirm Password** text boxes.

   Your password must be at least six, but fewer than twelve, alphanumeric characters.

4. When you are finished, click **Update** to update your password. The next time you log into RateManager, you will use the new password.

PROGRAM EXPLORER

Program Explorer serves as a central location for creating and managing programs and accessing program elements. The Program Explorer tab is available to all users. Access to the contents is determined by permissions. If there is a functionality or content that you need access to, please contact your system administrator.

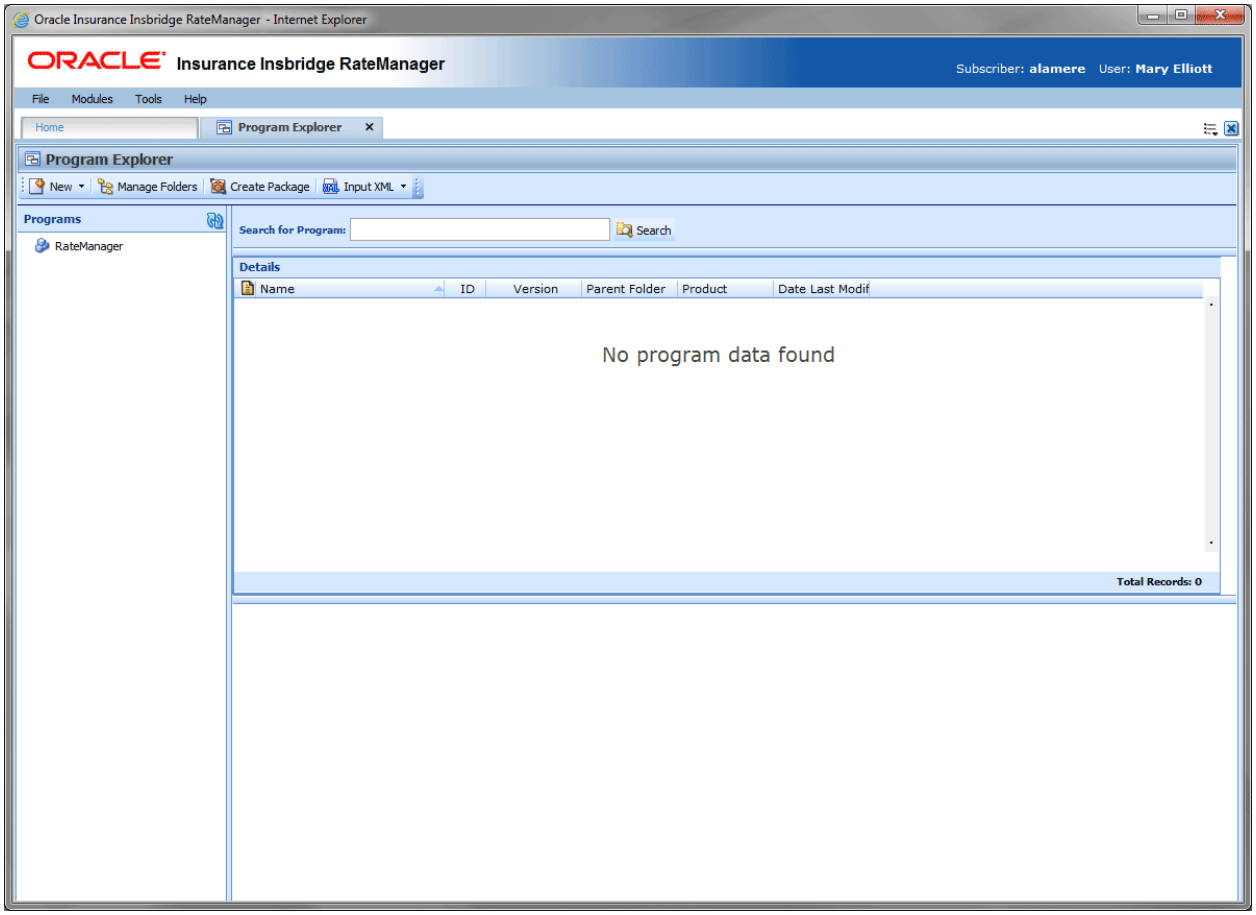


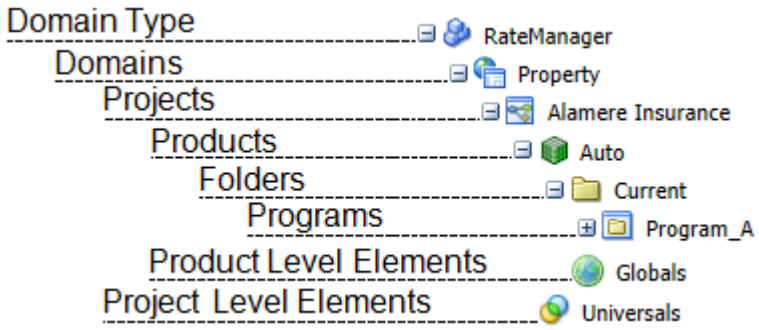
Figure 12 Program Explorer Without Content

Program Explorer Hierarchy

RateManager has an expansive and comprehensive structure where you can organize and define rating programs and rating elements in to one or more domains, projects, products, or folders. The classification of data and relationships can be as complex or as simple as you need, which allows you to create programs under a wider scope where elements can be shared at various levels or be unique to the program.

There are multiple levels where you can classify or manage data. For more information, see Chapter 4, Domains and Domain Types and Chapter 5, Projects and Products.

The basic structure must be in place prior to any programs being created.



- **Domain Type** – The top level is the domain type. A domain type designates the source of the material.
- **Domains** – Domains are top level or root containers for one or more projects. Depending on the structure you want and the shared content you create, you can have one or more domains. For example, you can have domains based on companies, products or territories.
- **Projects** – Projects are containers for one or more products. Like domains, you can create one or more projects to further classify programs and program elements. This level also holds universal categories and elements that can be used by any program in the project.
- **Products** – Products are line of business classifications that also can further define the project. Each project must have at least one product, but more products can be added if needed. This level also holds global elements that can be used by any program in the projects.
- **Folders** – Folders are the holders for one or more programs. Every product must have at least one folder but more can be added if needed.
- **Programs** – Programs are the rating programs. This level also holds local elements that are unique to the program.

Permissions Required

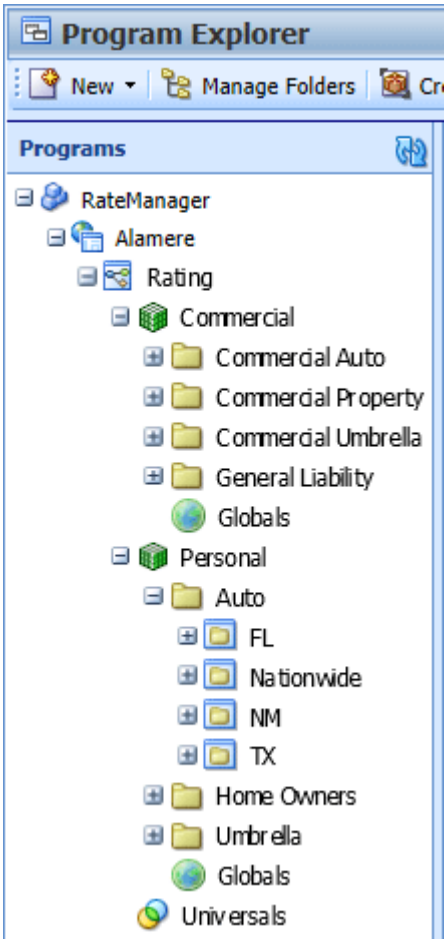
Everyone can see all domains, all projects and universals regardless of permissions. To work in these areas, you must have permissions. You must also have permissions to see and manage any products in the projects.

- **Product Management** – create, edit, activate, deactivate, disable and delete products.
- **Project Management** – create, edit and delete Other domain types. Create, edit, clear and delete domains. Create, edit, clear and delete projects.
- **Products** – create, edit and delete folders. Create, edit, and delete programs.
- **Universals** – create, edit and delete universal elements. Unless you have write permissions, you will only be able to view universals.

NOTE: *You cannot select products for a project unless you have both product and project permissions.*

Classification of Data

Prior to creating any programs, a classification structure must be in place under any domain type.



Simple, Single Level Setup Example

Under the RateManager domain type, a single domain has been created. In this example, the Alamere domain. This allows for all program and program elements to be in a controlled, uniform structure.

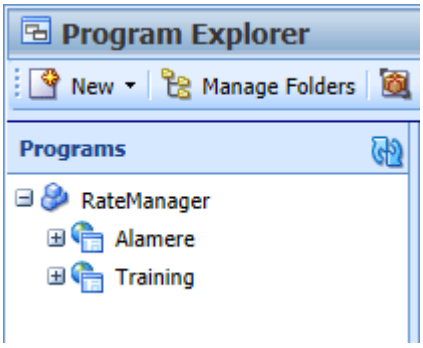
A single project has been created. In this example, Rating. This allows for a common category schema and common fields (input) to be used by every program in the project. These common elements are located in the universals area. A common schema also allows for easier management of program to program (P2P) callouts.

Two products have been added to the projects. In this example, Commercial and Personal. These products could be renamed standard products or custom products. Renaming an existing product with a broader name is for classification purposes. To visually make it easier to find the program needed.

Under each product, folders are created to reflect the lines of business used. These folders are also for classification use, to make it easier to locate a specific program.

Finally programs are then created that reflect the company program setup. In this example, it is by state or nationwide.

Creating Your Own Classification



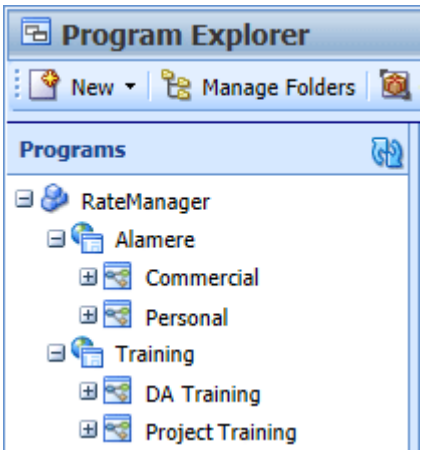
The first classification is domain types.

The RateManager domain type is most common.

The next classification is domains.

You may decide that you want two distinct and separate domains. One company level domain for RateManager users who create the rates (Alamere) and one for training (Training).

By keeping training separate, the risk of accidental changes to elements used in the creation of rates is reduced. This domain could even be populated with products that are only available to training. The Alamere domain also could be populated with products only available to RateManager users not in training.

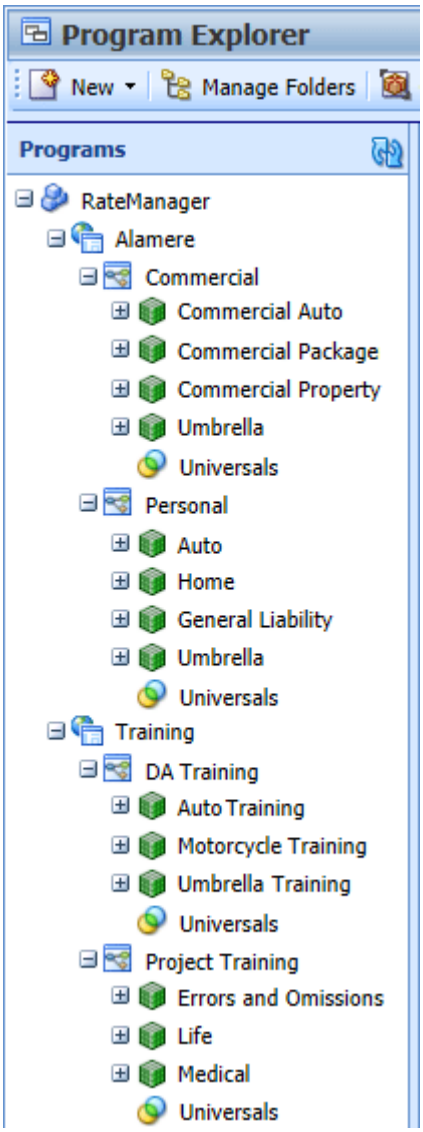


The next classification is projects.

You may decide that you want larger projects where products that share similar category structure and fields can be grouped.

Domains can hold one or more projects. Projects structure and content is determined by the user. RateManager does not place restrictions on what products or how many products can be included in a project.

In this example, two broad projects have been created for each domain.



Products are classified next.

After the projects have been created, the products can be added.

Products are line of business classifications.

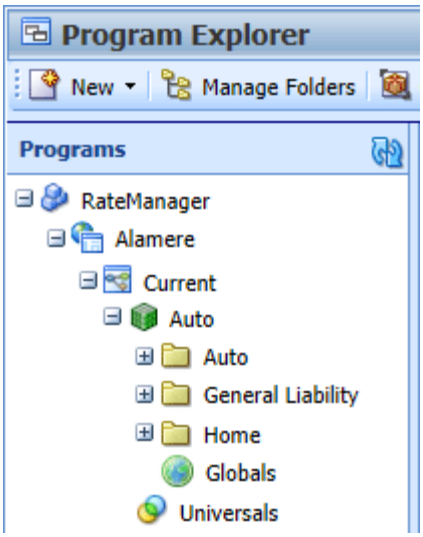
The setup of projects and products should be considered prior to any program creation. The setup can be simple with one project and a single product or complex with multiple projects and multiple products. The setup can be changed after the fact but it may be complicated and require some coordinated effort.

Things to consider may include:

- Is an auto product is required?
- How many program callouts are needed?
- Is there a need for multiple products?
- What categories and fields can be shared?
- Are there other variables that should be shared?
- Are there variables that need to be separate?

Universals

Every project has a Universals area that is displayed when a product is selected in a project. Universals hold the common elements that can be used in every product in the project.



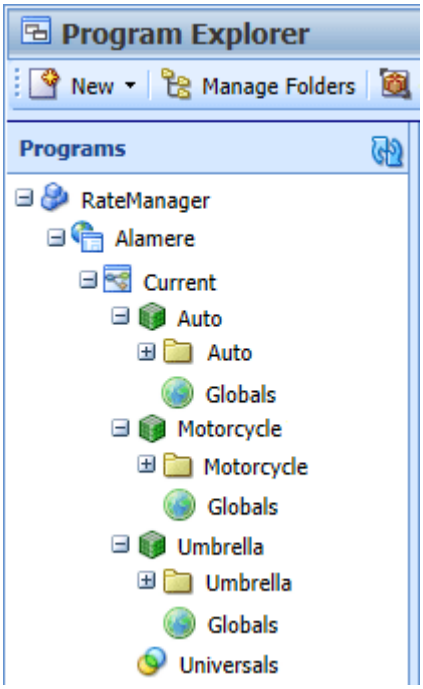
Option – Simple Product Structure

You can select one product and create multiple lines of business specific programs.

RateManager does not restrict the content of a product with the exception of driver assignment only being available in the Auto product. This allows for one product to be used for multiple kinds of programs.

For example, you can choose to only use the auto product and create programs for any line of business under the auto product.

This structure allows for a wider pool of shared categories and variables. Universal elements are shared by every program as well as all global elements.

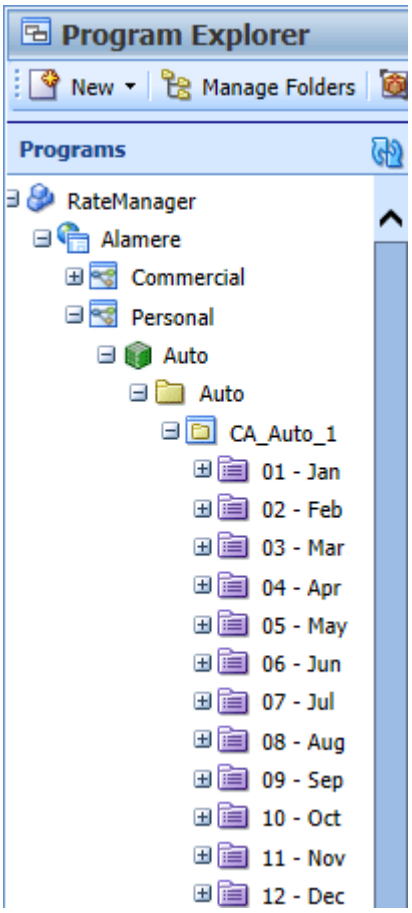


Option – Detailed Product Structure

You can use multiple products and create programs that are only for that line of business.

You can select as many products as you want. Each product under the project shares the universal elements.

This structure allows for clear separation of programs and products. Categories and fields are still shared but global elements are unique to the product.



Folders are the last classification.

The last classification is folders. Folders are holding locations for programs. Folders must be in place prior to creating a program.

The folder structure can be constructed in different ways. Either simple or detailed.

You can create a folder for every program, or create all programs under a single folder.

If you decide to have each line of business placed in its own folder, you can further define folders with optional program version folders.

Folders that are located under the same product share the same element structure. This allows for programs to be moved easily from one folder to another.

Once folders have been created, programs can be added.

Program Explorer Structure

The projects and products that you have permission to view are displayed on the left side of the tab. Under the domain are the universal elements including fields and categories. Under the products are folders, programs and globals that belong to that product. All program elements; variables, algorithms, sequencing, and output mapping can be found within the Program Details tab of any selected program.

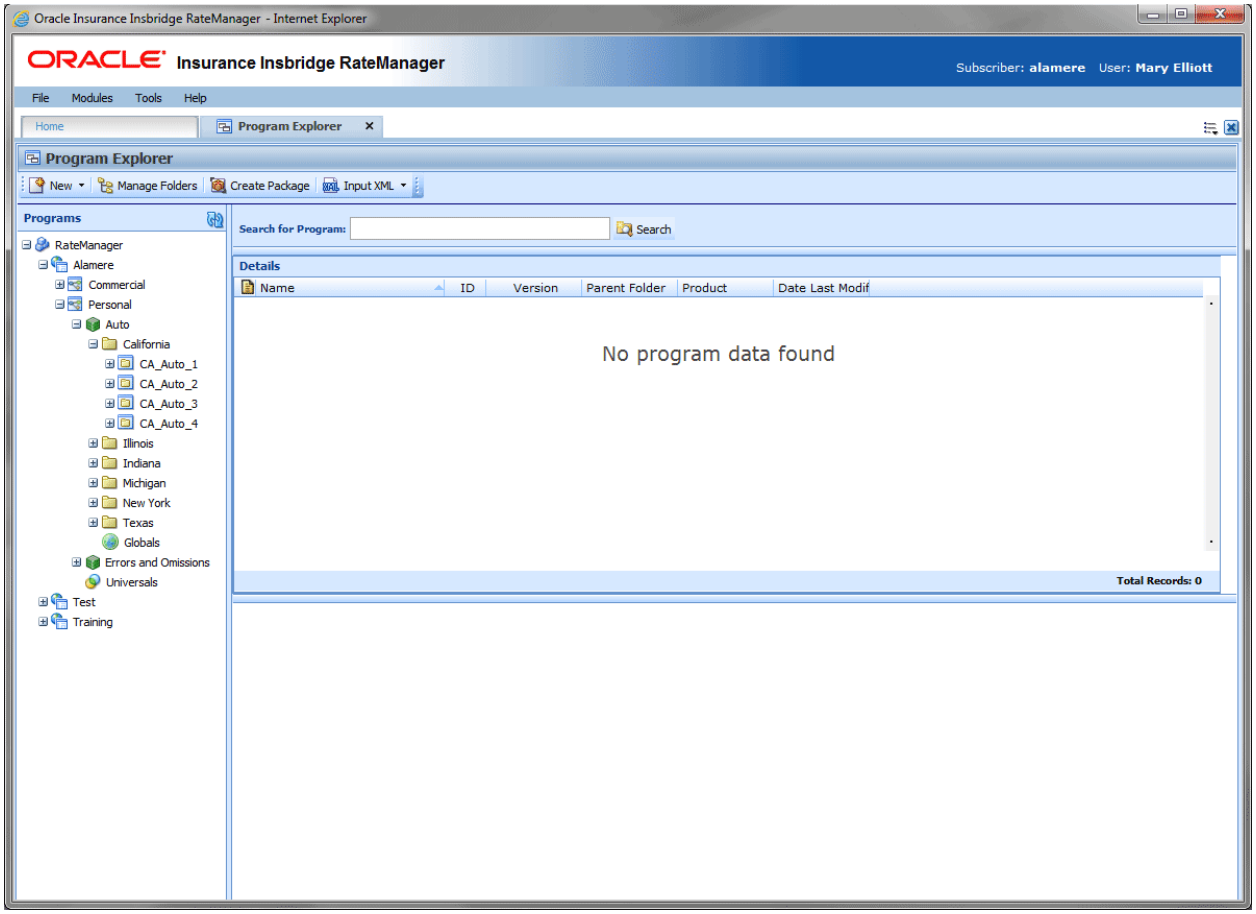


Figure 13 Program Explorer With Content

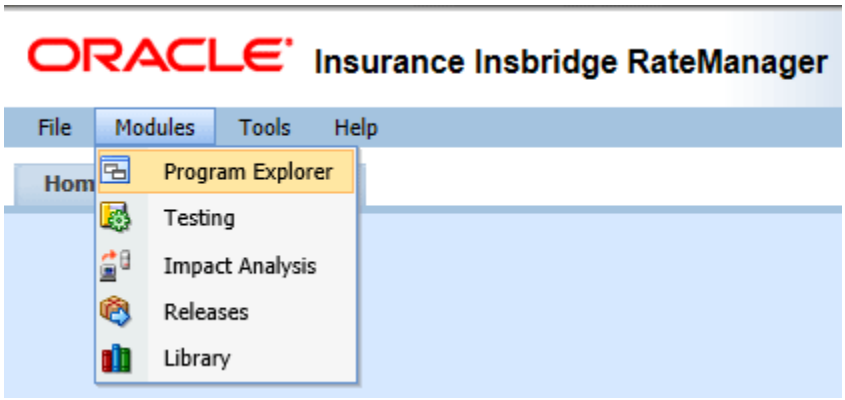
Program Explorer is divided into two areas:

- **Programs** – Programs is a navigation column on the left side of the tab. Programs displays the products you have rights to. As you expand the product, you can see the folders, and programs currently in the product.

- **Program Detail** – Program detail contains the details of the element you have selected from Programs. The item you select in the Programs area determines what information is displayed in the Program Detail area.
  - Domain types, domain and projects do not populate the Program Detail area.
  - Select the **products** to view the **folders**.
  - Select the **folders** to view the **programs**.
  - Select the **programs** to view the **program versions**.

Program elements are displayed in a separate tab from the program version. You also can select a program element from a snap on your Home tab.

Navigating to Program Explorer

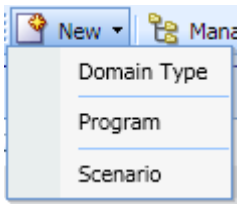
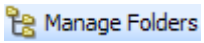
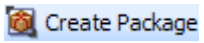
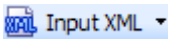


- You can get to Program Explorer from the top menu bar. Program Explorer is located in Modules→ Program Explorer.
- You also can click the Program Explorer Icon in the Control Panel on the Home tab.

PROGRAM EXPLORER FUNCTIONALITY

The **Menu Bar** is located at the top of the **tab**. The menu bar contains the functions you can perform on this tab as well as a search option.

Functions:

	**New:** Creates a new program, or a new domain type, or a new scenario. To create a new program, select the domain type→domain→project→product→folder where the new program should be created prior to clicking New Program. A new domain type is added to the listing in alphabetical order. Creating a new domain type requires domain admin permissions. Creating a new scenario requires scenario manager permissions.
	**Manage Folders:** Opens the Manage Program Folders function for the selected project→product.
	**Create Package:** Creates a package of the selected program version.
	**Input XML:** This option brings up the XML for: • **Program Version:** If you select a program version, the input XML for this program version is displayed. This option is available only if you select a program version. • **Project:** If you select the project, or the product→folder or product→folder→program, the product inputs XML is displayed.

Search:

The ***Search*** field on Program Explorer searches for a particular program. Select the product where you want to search. Type or paste the program name you are looking for and then click **Search**. Searches are at the product level. You cannot search one folder only. If you select a folder, the search is done at the product level.

As you type, matches populate the area below. The program name, program ID and folder location are listed. If the program you are looking for is there, you can click the match and your program is displayed.

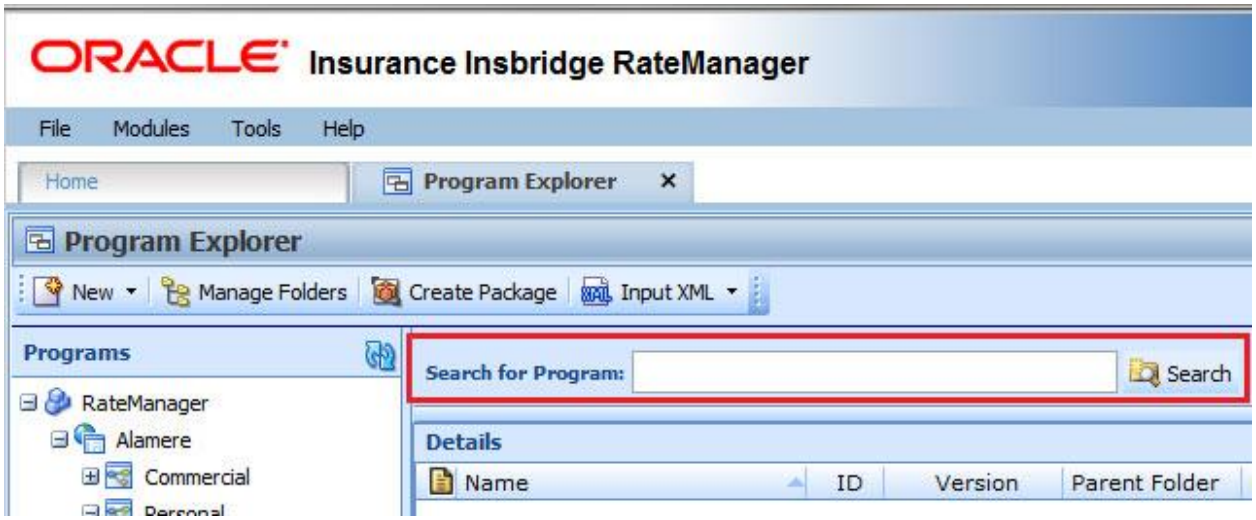


Figure 14 Program Explorer Search

PROGRAMS

Programs is the navigation column on the left side of the Program Explorer. Icons are displayed as the levels are expanded. Project, Folders and Programs are displayed after they are created.

PROGRAMS SCREEN DETAILS

Screen details are labeled with the current name assigned to them.

	RateManager Domain Type: A top level domain type. The RateManager domain type is the only default item on Program Explorer and cannot be deleted.
	Other Domain Type: A top level domain type.
	Domain: Domains are top level or root container for one or more projects.
	Projects: Projects are containers for one or more products.
	Products: Products are line of business classifications.
	Folders: Folders are containers for one or more programs.
	Program Version Folders: This optional folder is displayed under every program in the folder. Individual program versions can be sorted in to program version folders.
	Programs: Rating programs.
	Globals: Globals are elements available to every program in the product.
	Universals: Universals are available to every program in the project.
	Scenario: Scenarios are available to users with scenario permissions only.
	Scenario Domain: Scenario Domains are available to users with scenario permissions only.
	Refresh: Refreshes the screen with the current information.

Right Click Menu Options:

In the Programs area, select the field and right click to view a menu. Program version folders do not have a right click menu.

	RateManager Domain Type: Has one option available to users with Domain Management rights. • **New Domain:** Create a new domain.
	Other Domain Type: Has three options available to users with Domain Management rights. • **New Domain:** Create a new domain. • **Edit Domain Type:** Allows for the editing of the domain type name. • **Delete Domain Type:** Deletes the selected domain type.
	Domain: Has four options available to users with Domain Management rights. • **New Project:** Create a new project. • **Edit Domain:** Allows for the editing of the domain name. • **Clear Domain:** Clears the domain of all content.

	• **Delete Domain:** Deletes the selected domain.
	Projects: Has five options available to users with Domain Management rights. • **Map Custom XML:** Opens the Map Custom XML for the global fields in the project. • **Select Products:** Opens the Select Products window. • **Move Project:** Moves the project from the current domain to another domain. • **Clear Project:** Removes all contents, programs and data, from the project. • **Delete Project:** Deletes the entire project.
	Products: Has four options available to users with Domain Management rights. • **New Folder:** Create a new folder. • **Create Versioning Package:** Create a global versioning package. • **Global Versioning Report:** Create a global versioning report. • **Category Visibility:** Allows for categories to be available or disabled for the product.
	Folders: Has four options available to users with Product rights. • **New Program:** Creates a new program in this folder. • **New Folder:** Opens the New Folder screen where you can create a new folder. • **Edit Folder:** Opens the Edit folder screen where you can edit this folder. • **Delete Folder:** Deletes the selected folder.
	Program Version Folders: No right click menu.
	Programs: Has six options available to users with Product rights. • **Program Properties:** Brings up the Edit Program screen where you can make edits to the versioning criteria and other program information. • **Move Program:** Moves a program from one folder to another within the same project. • **Delete Program:** Deletes the selected program. • **Map Callouts:** Map callouts for the selected program. • **Program Difference Report:** Brings up the Program Difference Wizard. • **Global Versioning Report:** Create a global versioning report.
	Global: One option, Open. Opens the Global Details tab for the product.
	Universal: One option, Open. Opens the Universal tab for the project.
	Scenario: Has one option available to users with Scenario Management rights. • **New Scenario:** Create a new scenario.
	Scenario Domain: Has three options available to users with Scenario Management rights. • **Select Programs:** Opens the Select Products window. • **Edit Scenario:** Allows for the editing of the scenario name. • **Delete Scenario:** Deletes the selected scenario.

PROGRAM DETAILS

The Program Details area is where the details of the element you have selected from Programs is displayed. The item you select in the Program Folders area determines what information is displayed in the Program Details area.

PROGRAM DETAILS SCREEN DETAILS

The results can be sorted by column. To change the order, click the column you want to sort by. The sort icon indicates the column that is determining the sort order.

SELECTION	DISPLAYS
	Domain types, domains, projects and scenarios do not display anything in Program Details.
Products	Information for the folders that were created under the selected product. **Icon:** **Name:** The name of the folder. **ID:** Not used for folders. **Version:** Not used for folders. **Parent Folder:** Displays the pathway of the folder. **Product:** The product where the folder is located. **Date Last Modified:** The date the folder was last modified.
Folder	Information for the programs that were created under the selected folder. **Icon:** **Name:** The name of the program. **ID:** The program ID number. **Version:** Not used for programs. **Country:** The country this program was created for. **State/Province:** The state/province this program was created for. **Parent Folder:** The folder where this program resides. **Product:** The product where the program is located. **Date Last Modified:** The date the program was last modified.
Program	The program versions for the selected program. **Icon:** **Name:** The name of the program. **ID:** The program ID number. **Version:** The program version number. **Version Name:** The program version name. **Lock:** An administrator can lock the program to prevent it from being changed or deleted.

	Parent Folder: The folder where this program version resides. **Product:** The product where the program version is located. **Date Last Modified:** The date this program version was last modified. **Last Packaged On:** The date and time the last RateManager package was created for the version. **Note:** An icon indicates that this program version has a note. **Version Folder:** If this program version has been placed in a program version folder, the program version folder name is listed.
Program Version Folder	The program versions for the selected program. **Icon:** **Name:** The name of the program. **ID:** The program ID number. **Version:** The program version number. **Version Name:** The program version name. **Lock:** An administrator can lock the program to prevent it from being changed or deleted. **Parent Folder:** The folder where this program version resides. **Product:** The product where the program version is located. **Date Last Modified:** The date this program version was last modified. **Last Packaged On:** The date and time the last RateManager package was created for the version. **Note:** An icon indicates that this program version has a note.
Global	No change.
Universal	No change.

Right Click Menu Options:

The right click menu options are available on most elements displayed in the Program Detail area. The menu depends upon the element selected and the rights you have.

SELECTION	DISPLAYS
	Domain types, domains, projects, and scenarios do not display anything in Program Details.
Folder	When Products are selected, information for the folders that were created under the selected product is displayed. Folders have two options: • **Edit Folder:** Opens the Edit Folders screen where you can edit the folder. • **Delete Folder:** Deletes the selected folder.
Program and Program Version Folders	When folders are selected, information for the programs that were created under the selected folder is displayed. Programs have seven options: • **Program Properties:** Opens the Edit Program screen where you can make edits to the versioning criteria and other program information. • **Move Program:** Opens the Move Program screen where you can move a program within the project. • **Delete Program:** Deletes the selected program. • **Map Callouts:** Opens the Callout Mapping screen where you can enter the mappings for program callouts or library callouts. • **Program Difference Report:** Create the program difference report. • **Global Versioning Report:** Create a global versioning report. • **See Program Versions:** Refreshes the tab with the program versions.
Program Version	When programs are selected, the program versions are displayed. Program versions have 12 options: • **New Version:** Creates a new program version from the selected program version. • **Open:** Opens the Program Details tab for this program version. • **Edit Versioning Data:** Opens the Edit Program Version screen where you can make edits to the versioning criteria and other program information. • **Copy:** Creates a copy of the program version. • **Delete:** Deletes the program version. • **Test Program:** Opens Test Program, where you can test and debug a program within RateManager. • **Map Custom XML:** Opens the Custom XML Mapping screen for this program version. • **Notes:** Enter, edit and view notes to the program version. • **Program Version Report:** Create a program version report. • **Global Versioning Report:** Create a global versioning report. • **Unassign From Program Version Folder:** removes a program version from the program version folder where it was assigned. • **Assign To Program Version Folder:** Opens the Assign Version to Program Version Folder screen. Here you can assign a program version to a folder.
Global	No change.

Universal	No change.

Program Creation

Creating and packaging a program is done in Program Explorer. Program Explorer is open to all users for viewing, but permissions are required to create and manage domains, projects, folders, programs and program elements.

Listed are the steps and permissions that are required to create a new program in a new RateManager instance.

1. **Create a domain** – Need Project Management rights.
2. **Create project** – Need Project Management rights.
3. **Select products** – Need Project Management rights and Product Management rights. If permissions to the product need to be set, admin rights are needed for Group Management.
4. **Create folder** – Need Product specific permission.
5. **Create program**– Need Product specific permission.
6. **Create categories** – Need Universal rights. If category visibility needs to be set, Project Management right are required.
7. **Create fields** – Need Universal rights.
8. **Create global results** – Need Product specific permission.
9. **Create local/global results algorithm** – Need Product specific permission.
10. **Create sequence** – Need Product specific permission.
11. **Create output mapping** – Need Product specific permission.
12. **Package program** – Need Product specific permission.

Once the program is packaged, it is ready for testing in Test Case Editor.

SCENARIO SANDBOX

The Scenario domain type serves as a sandbox for users to test table changes. A Scenario Sandbox (scenarios) is a secure environment that holds copies of RateManager programs and copies of tables. The purpose of a scenario is to test table changes within a program without affecting the original program or tables. Scenarios are only visible to users with any scenario permissions. If you do not have scenario permissions, the scenario domain type will not be displayed.

Most often, a scenario user will only have access to the Scenario domain type and not the RateManager domain type. When a scenario user logs in, the Program Explorer view will contain the Scenario domain type and any scenarios where access has been granted.

The Scenarios domain type follows the structure of RateManager and Other domain types. There are projects, where programs are added, and where tables are added. Universal tables are also available.

In a Scenario domain type, table data, table criteria and default values are editable. Other variables and program elements are not available. Packages can be created and tested.

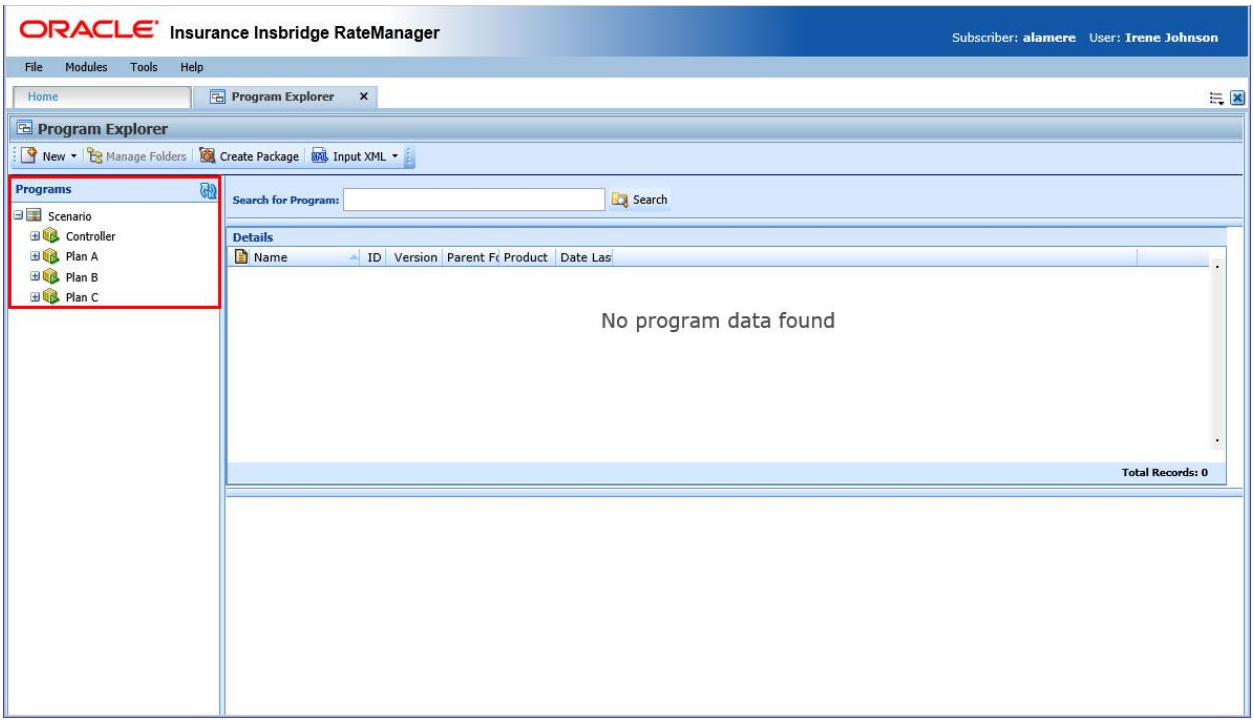


Figure 15 Scenario User View

Permissions must be granted to work in this area. If you cannot access scenario, please contact your system administrator to setup permissions.

Subversions and Subrevisions

Scenarios hold copies of program versions called subversions. Subversion have unique identifiers that distinguish the copy from the original and from other copies. Programs can be added to multiple scenarios. If the same program is in multiple scenarios, the subversion is unique for each one.

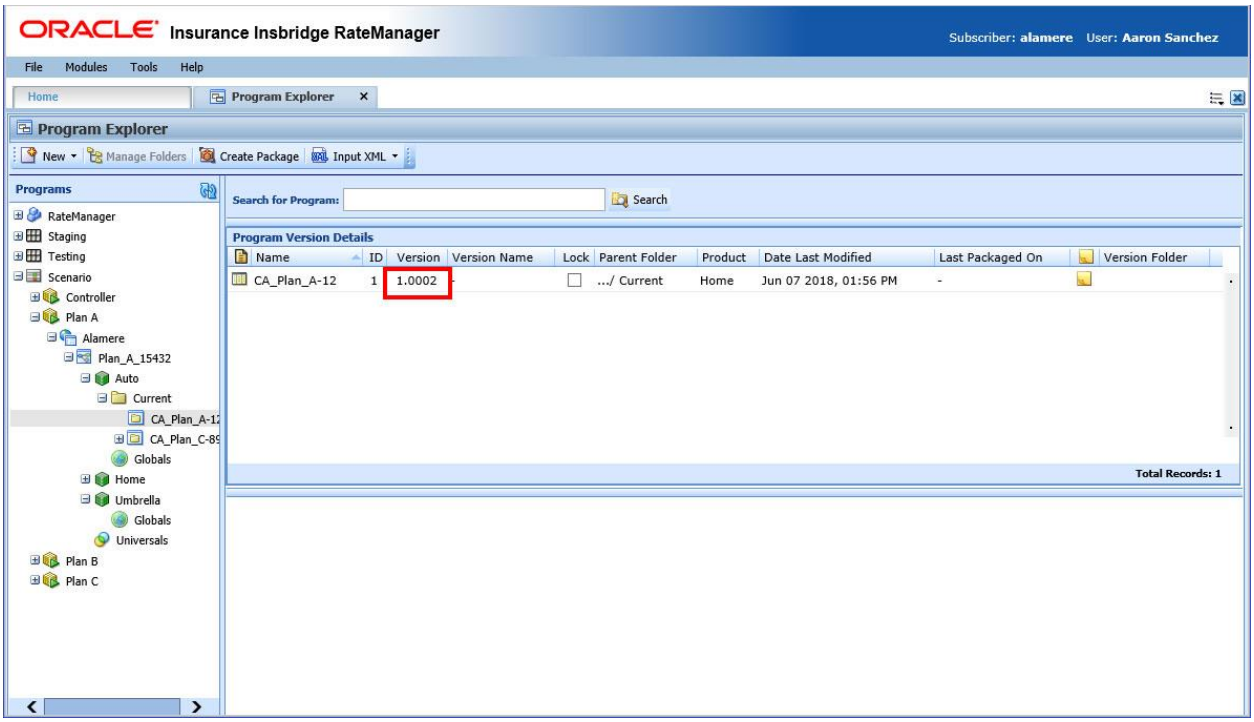


Figure 16 Scenario Program Subversion

The tables that are added to a subversion are called subrevisions that users with permissions can edit. A subrevision is a unique identifier for the table that distinguishes the table from the original and from other copies. Tables can be added to multiple scenarios. If the same table is in multiple scenarios, the subversion is unique for each one.

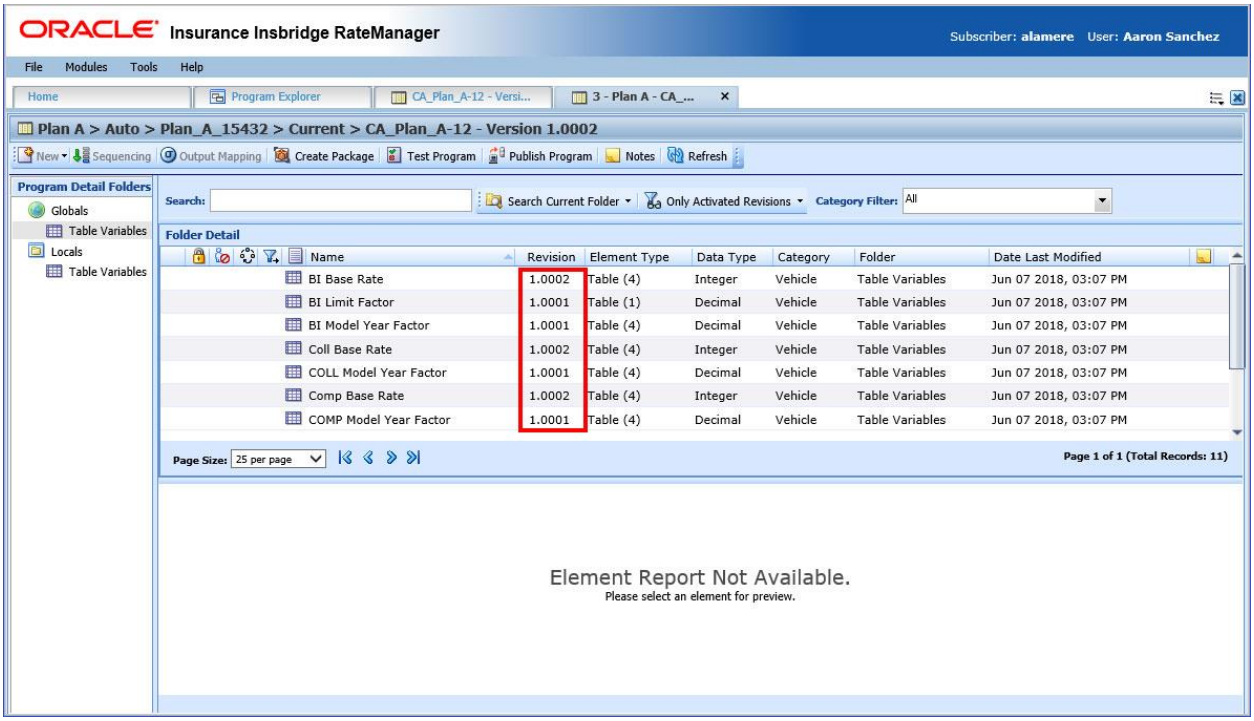


Figure 17 Scenario Table Subrevisions

Subversions and subrevisions are copies and do not affect the original programs or tables.

Scenario Icons

There are icons that are unique to Scenarios.

Icon	Description
	Scenario Domain Type
	Scenario Update Required
	Open Scenario
	Locked Scenario
	Published Program

Scenario Update Required

Scenarios from a previous release may need to be updated. If the scenario has update icon, the scenario has to be converted to the newer format, Sandbox Scenario. Converting a scenario will take it from a previous version to the current version and allow for the scenario to be worked on. You will not be able to work in a scenario that requires an update. Scenario manager permissions are required to convert an ol scenario.

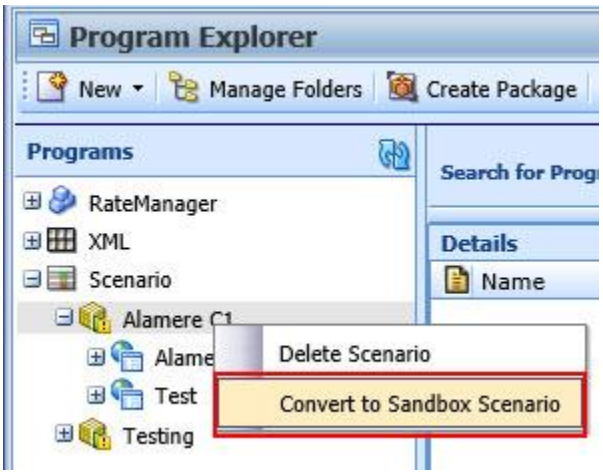


Figure 18 Scenarios Update Required

1. Select the scenario that needs updating and right click.
2. Select Convert to Sandbox Scenario.
3. A confirmation message is displayed. Click OK to continue. Click Cancel to stop.
4. The scenario will be converted. A success or fail message is presented. Scenarios that have been successfully converted will now have an Open Scenario icon.

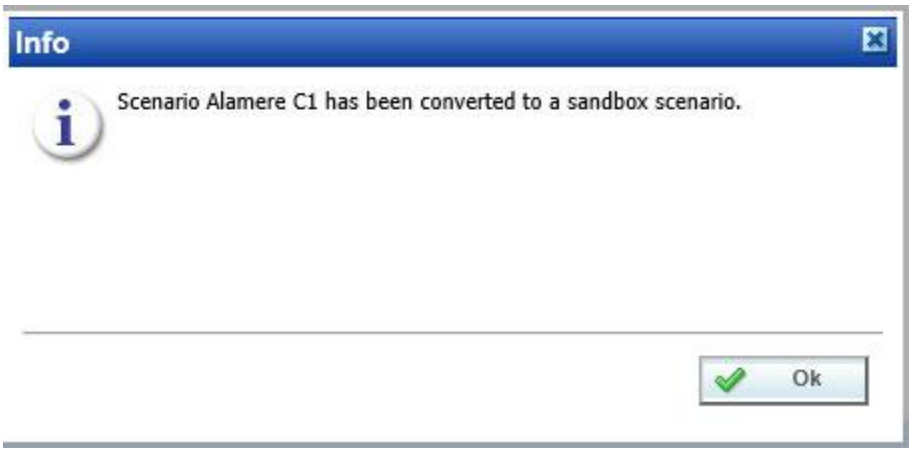


Figure 19 Successful Scenario Conversion

Scenarios only need to be converted once. After the scenario is converted, the update icon is replaced with the open icon.

Scenario Permissions

There are permissions that must be granted in RateManager before users can start working with scenarios. You must have admin permissions to perform these tasks.
There are two set of permissions that can be applied;

- **Scenario Manager Permissions:** This user creates scenarios, adds program versions, and adds tables.
- **Scenario Permissions:** This user edits table data and default values.

Separate permissions keep the users from crossing over from a scenario view to a standard RateManager user.

Scenario Manager Permissions

To allow for the complete setup of scenarios, the scenario manager needs at least two sets of permissions. Additional permissions may be needed, depending on usage. Permissions are required if universals are to be used. To test updates or delete programs, the user needs Program Management for the product. Product permissions are required to edit and delete tables.

Set 1: Scenario Manager Admin permissions are required to create, edit, and delete scenarios. Permissions must be assigned at the group level. These permissions apply to all users assigned to the group.

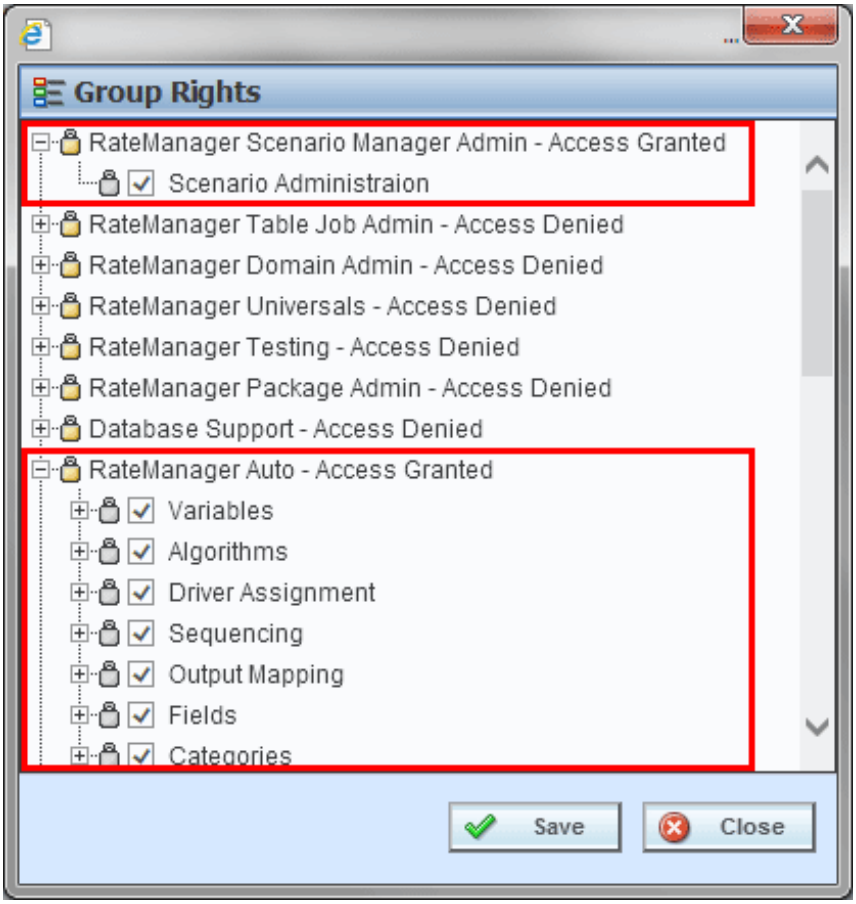


Figure 20 Group Rights for Scenario Managers

Set 2: Product permissions for each of the products with program versions and tables that are to be included in the scenario are required. Product permissions allow for program versions and tables to be added to the scenario. The product permissions can be read only.

Set 3: OPTIONAL - Permissions are required if universals are to be used. To test updates or delete programs, the user needs Program Management for the product. Product permissions are required to edit and delete tables.

Admin permissions are required to edit Group Rights.

If scenario manager permissions are not given, scenarios will not be presented on the Program Explorer tab and the New Scenario option in the top bar menu will be grayed out. If product permissions are not given, an access denied message is displayed.

Scenario User Permissions

This user edits table data and default values. Scenario user permissions allow for only tables to be edited. No other program elements are available. Scenario user permissions are per product and are needed for each product where the user needs to work.

To allow for the complete setup of scenarios, the scenario user needs at least two sets of permissions. A third set may be needed depending on usage.

Scenario user permissions are product wide. Users can work in the product in any scenario.

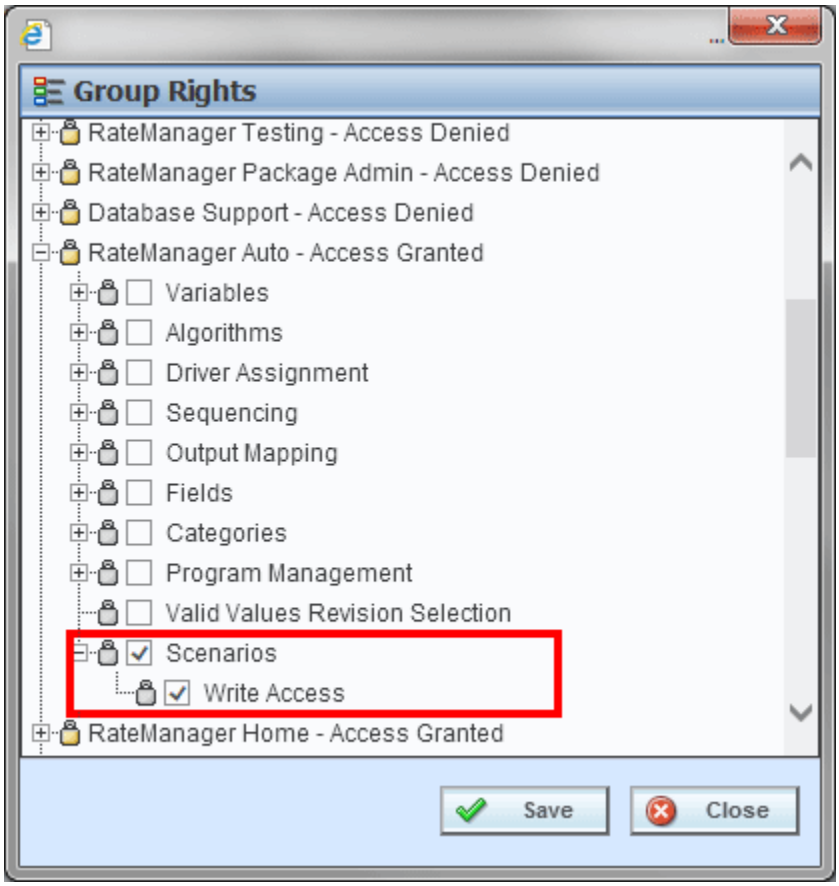


Figure 21 Group Rights for Scenario Users

Set 1: Scenario permissions for each of the products with the tables that are to be edited are required. Scenario permissions must be added in each product where the user needs to work.

Set 2: Excel table imports and exports require Table Job Admin – Import/Export permissions.

Set 3: OPTIONAL - Additional permissions may be needed. Permissions are required if universals are to be used. To test updates or delete programs, the user needs Program Management for the product. Product permissions are required to edit and delete tables.

Admin permissions are required to edit Group Rights.

If scenario permissions are not granted, scenarios will not be presented on the Program Explorer tab. If Table Job permissions are not granted, users will not be able to import or export Excel files.

SCENARIO CREATION PROCESS

You must have Scenario Manager Permissions to create, edit, and delete scenarios. Scenarios can be setup in any configuration. You can have one or more scenarios, with one or more programs, with one or more tables. For example, you can create one scenario with one program and multiple tables. Or one scenario with multiple programs and tables. Or multiple scenarios, each with a different program and different tables. Or add the same program to two different scenarios with different tables in each one. There is no limit to the number of scenarios you can create or the combinations of programs and tables that you can use.

Creating a scenario is a three part process.

- Create a Scenario
- Add the Program Versions
- Add the Tables

After the scenario has been setup, the scenario user can begin work.

When work is finished:

- Programs can be published
- Scenarios that are no longer needed can be deleted

Notes for Scenarios

A scenario is a sandbox where users can test the effects of table changes to a program. After work has been completed, the program can be deleted from the scenario if it is no longer needed or published from the Scenario sandbox to the RateManager domain type. Publishing has two options: to create a new program version or to merge the changes from the scenario program to the program in the RateManager domain type.

When any program in the scenario is published, that scenario is locked. Locking stops work from being done to any program in the scenario, preventing unintended changes from being made. A scenario can be unlocked however; unlocking has the potential to overwrite the work you just published.

When creating a scenario, you can start with a single program or a small number of programs in a scenario, which limits the number of programs affected if the scenario is locked. Likewise, it may be prudent to not duplicate programs or tables in multiple scenarios to limit the potential of overwriting your work.

Criteria can be added to a table variable in a scenario. If a new field or result or other variable is needed, it must be added prior to the program being added to the scenario. Field, result and other variables cannot be created in any scenario sandbox.

Creating a Scenario

1. If no scenarios are listed, click New->Scenario in the top bar menu.

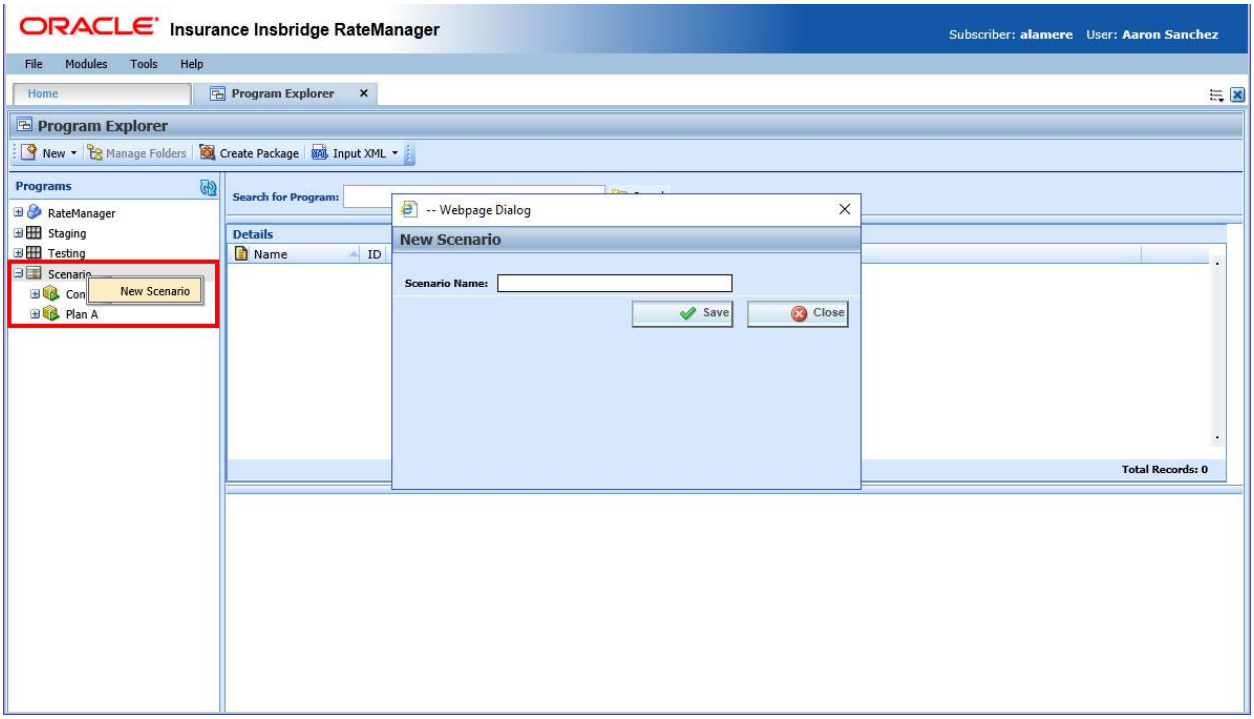


Figure 22 Right Click menu for Creating Scenarios

2. If scenarios are listed, a right click menu is available on the scenario domain type that allows for a new scenario to be created.

3. Enter in a Scenario name and click Save. Click Close to cancel scenario creation.

4. The screen refreshes and the scenario is listed.

Adding Programs to a Scenario

The choice of programs and tables is determined by the permissions the user has. If the scenario manager has permissions to the product, programs in the product can be added to a scenario. If the product is not listed, permissions have not been granted.

Only one version of a program can be added to a scenario. When you select a program version, the domain, project, product, folder, and program version folder where the program version resides is also listed in the scenario.

Unlocked programs can be added from the RateManager domain type only. Locked programs cannot be added to a scenario.

You can add an incomplete program to a scenario. This is a program that may not have a sequence or an output group created. Tables can be added to and edited in this subversion. You cannot package or test in scenario but you can publish and complete the program in the RateManager domain type.

On a subversion program, versioning criteria is open for editing. Everything else is locked.

1. Select the scenario where you want to add one or more program version.

2. Right click and choose Select Programs. The Select Programs screen is displayed. The choice of programs is determined by the permissions you have.

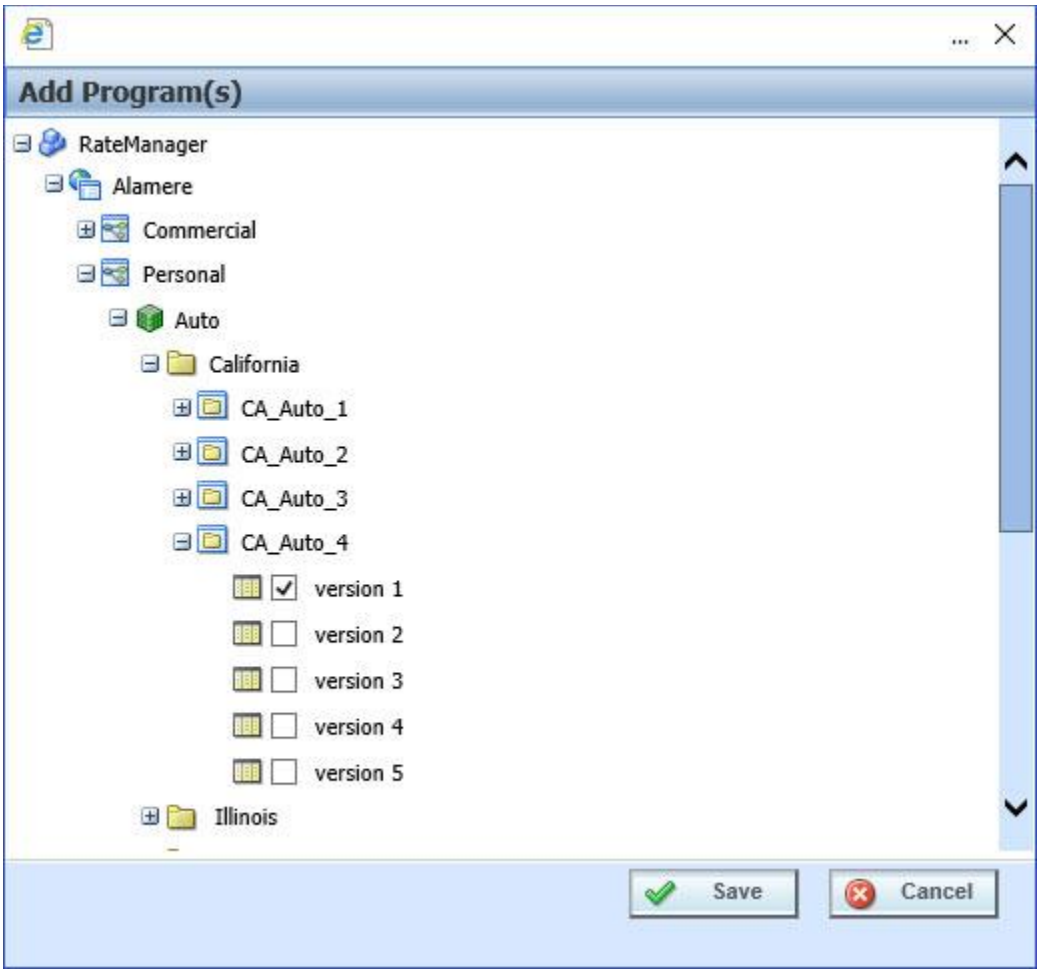


Figure 23 Adding Program to a Scenario

3. Expand the products and programs to select the program versions. Only one version of the program can be selected.

4. After selecting, click Save. The screen refreshes. The tree structure for the selected program version is also listed in the scenario.

You can add programs to an unlocked scenario at any time. If a program has been added to a scenario, there will be a check mark in front and other program versions will be unavailable.

Locked Programs

If there is no checkbox next to the program version, then the program version is locked and not available for inclusion in a scenario. For example, version 1 of Universal Coverage does not have a checkbox next to it. This program is locked and cannot be added.

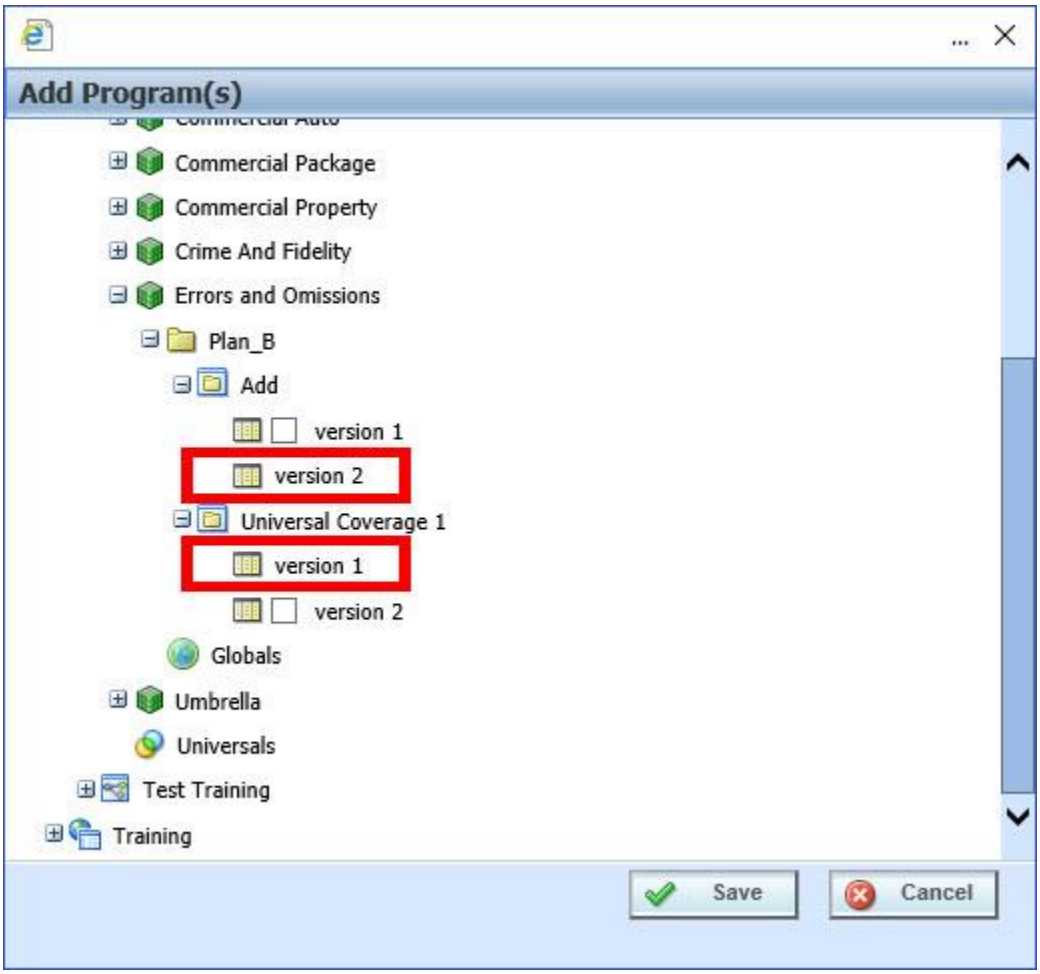


Figure 24 Locked Program Unavailabe to a Scenario

Removing a Program

If a program is no longer required, it can be removed. Removing a program version from a scenario does not remove the program from the RateManager domain type.

If any tables are associated with the program, and the program is removed, then the tables will no longer be available in the scenario.

1. Select the scenario where you want to remove a program.

2. Navigate to the program you want to delete. In the Program Version Details screen, right click the program and select Delete. Permissions are required to delete programs. If the Delete option is grayed out, you do not have permissions.

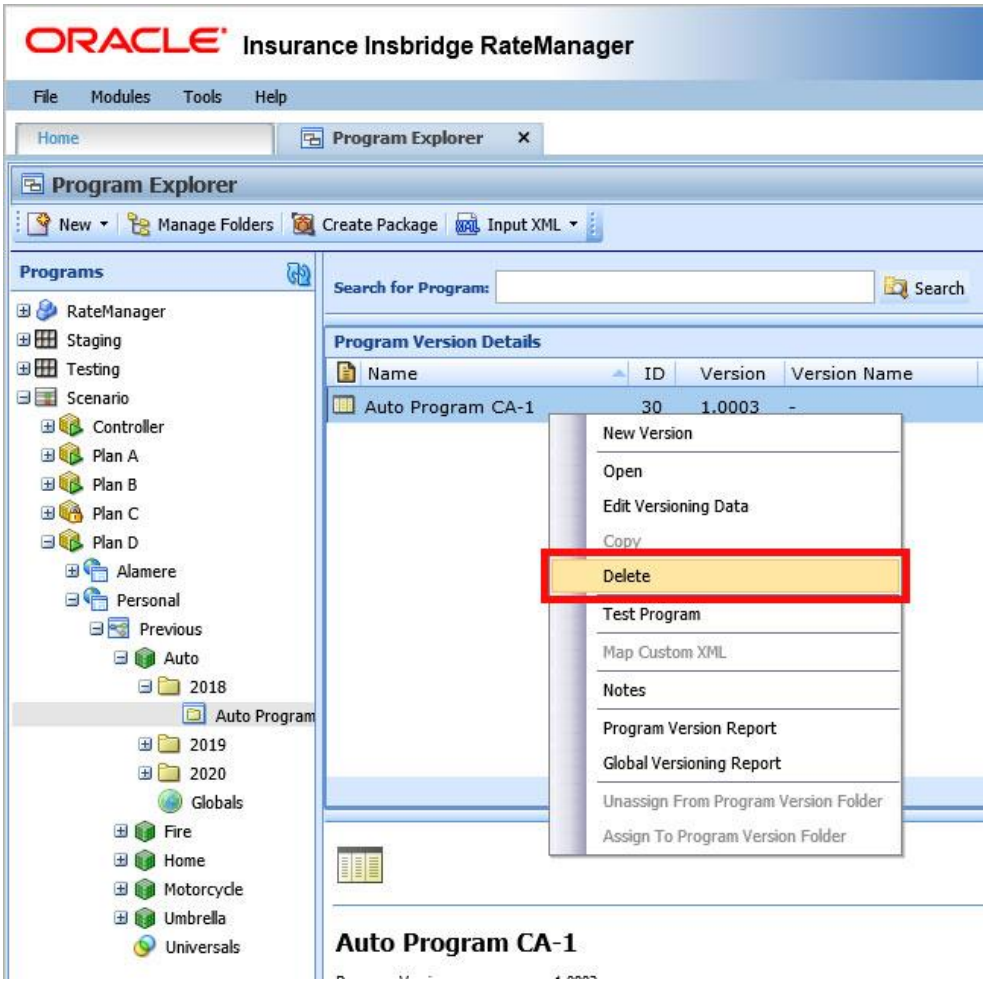


Figure 25 Removing a Program from a Scenario

3. Two confirmation message are displayed. Click OK to continue. Click Cancel to stop.

4. The program will be removed from the scenario only.

Adding Tables to a Scenario

The choice of programs and tables is determined by the permissions you have. If the scenario manager has permissions to the product, programs can be accessed and tables can be added to a scenario. If the product is not listed, permissions have not been granted.

Adding a table is adding an exact copy to the scenario where the default value, table data and criteria can be edited. Linked variables and advanced options cannot be added.

Tables can be added one at a time or in bulk. Globals are added separately from locals. Universals are added separtely as well. Only the active revision is added. All linked variables are added when a table with linked variables is added. The same tables can be added to one or more scenarios. When the same table is added to multiple scenarios, the subrevision number will be different.

1. In the Scenario Domain Type, open the program where the tables you want to add are located.

2. In the Program Details tab, select either global or local tables and right click.

3. Click Add to Scenario. A separate screen is displayed.

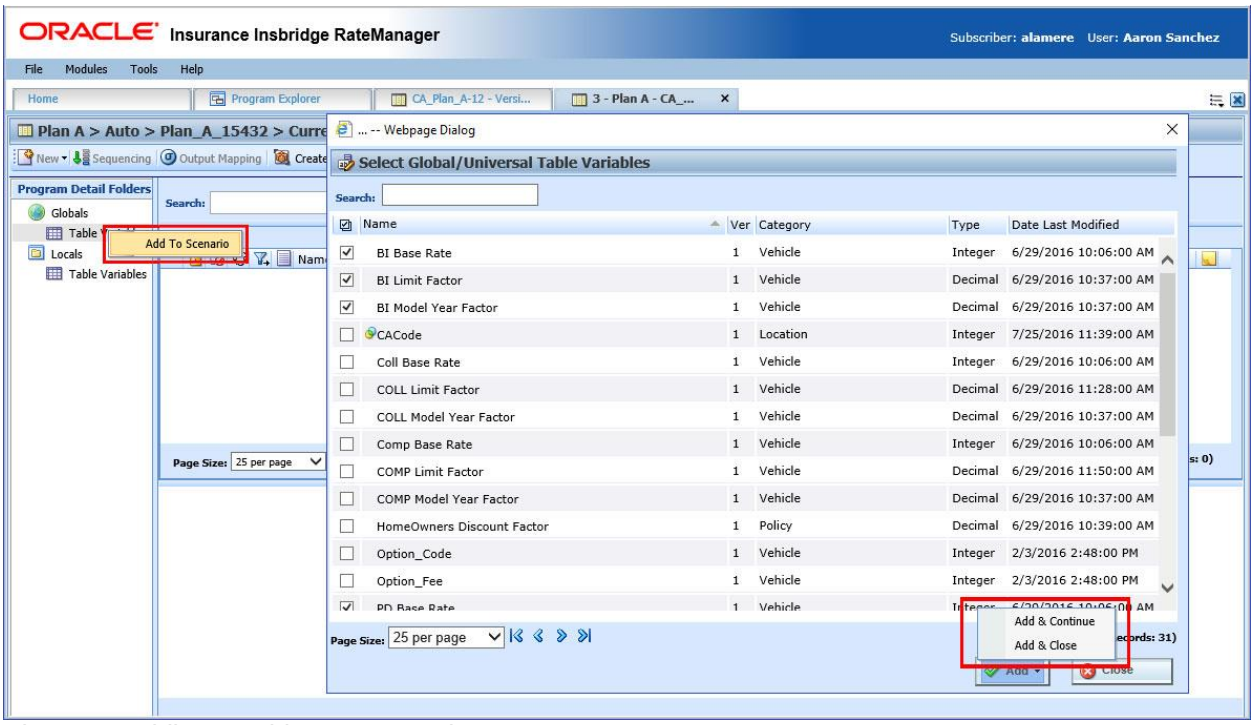


Figure 26 Adding a Table to a Scenario

4. Check the box in front the table you want to add. You can select one or more tables.

5. Click Add and choose if you want to Add and Continue or Add and Close. Add and Continue will add the tables and let you continue adding tables. Add and Close will add the tables and close the window.

6. The selected tables are now listed.

7. Repeat process for remaining tables.

Removing a Single Table from a Scenario

Single tables can be removed from a scenario at any time. Removing a table from a scenario does not remove the table from the RateManager domain type. Linked variables cannot be deleted from the Program Details tab.

1. In the Scenario Domain Type, open the program where the tables you want to delete are located.
2. In the Program Details tab, select the table you want to delete and right click.
3. Click Delete.

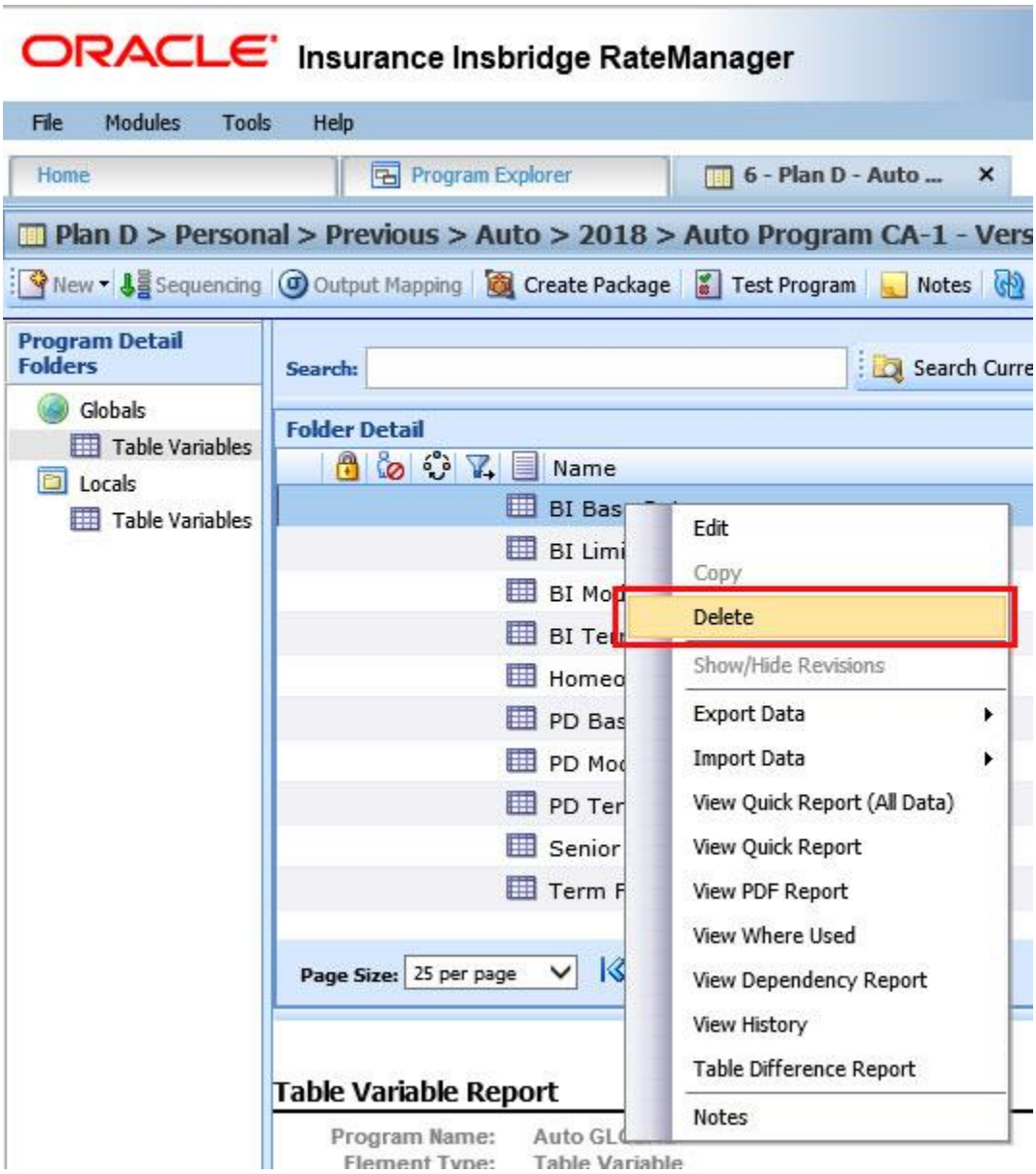


Figure 27 Removing a Table from a Scenario

4. A warning message is displayed. Click OK to delete, click Cancel to stop.

Removing a Linked Variable Table from a Scenario

Linked variables tables can be removed from a scenario. Removing a table from a scenario does not remove the table from the RateManager domain type.

1. In the Scenario Domain Type, open the program where the tables you want to delete are located.
2. In the Program Details tab, open the table where you want to delete a linked variable.
3. Select the linked variable you want to delete and click Delete in the top bar menu.

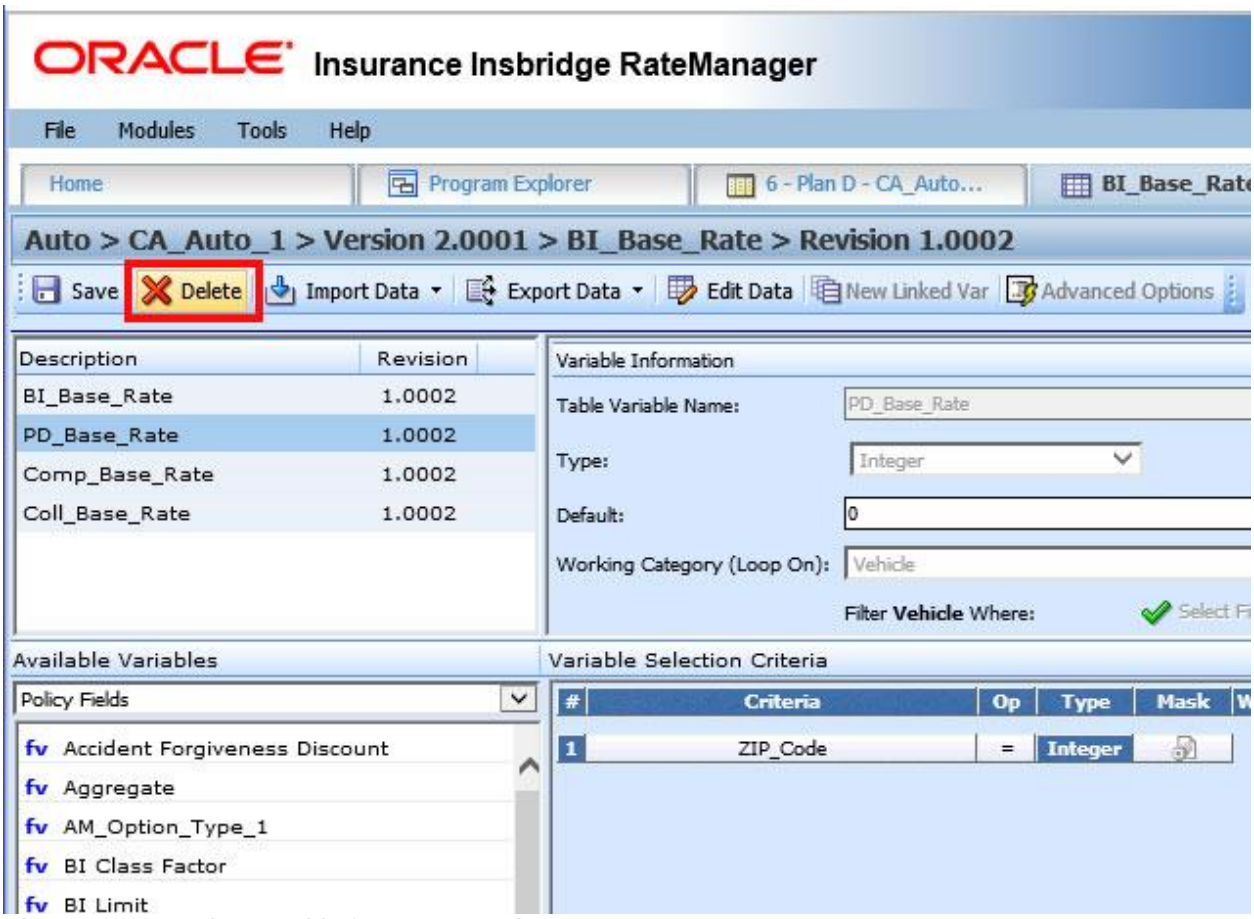


Figure 28 Removing a Table from a Scenario

4. A warning message is displayed. Click OK to delete, click Cancel to stop.

When you publish, the deleted linked variable will be removed from the RateManager domain type.

Using Where Used to Check Status

Tables must be removed from a scenario prior to being deleted in the RateManager domain type. Tables that are being used in a scenario cannot be deleted from the RateManager domain type until they are removed.

To check if a table is in a scenario, use the Where Used Option on the right click menu of the table.

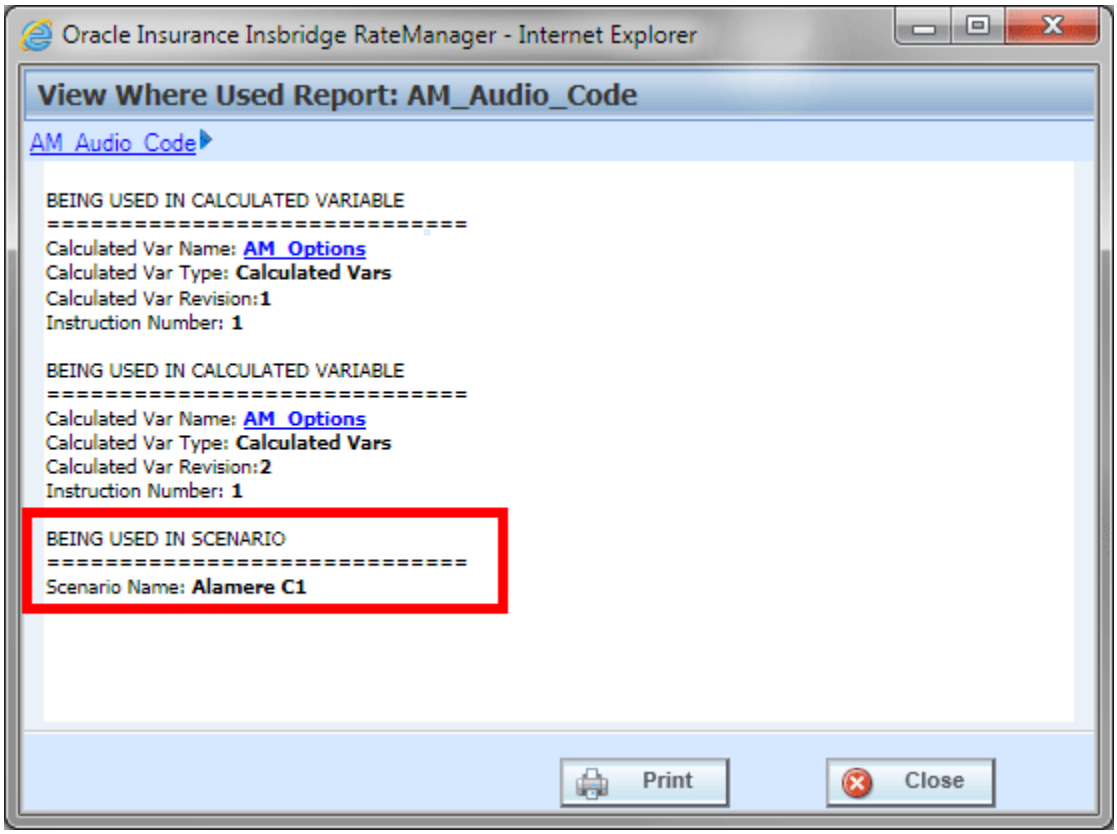


Figure 29 Using View Where Used to Check Table Status

Editing a Scenario

You must have Scenario Manager Permissions to create, edit and delete scenarios. Only the name of the scenario can be edited. To edit the list of programs contained in the scenario, use the Select Programs option. To edit the list of tables available in the scenario, please see Adding Table or Removing Tables.

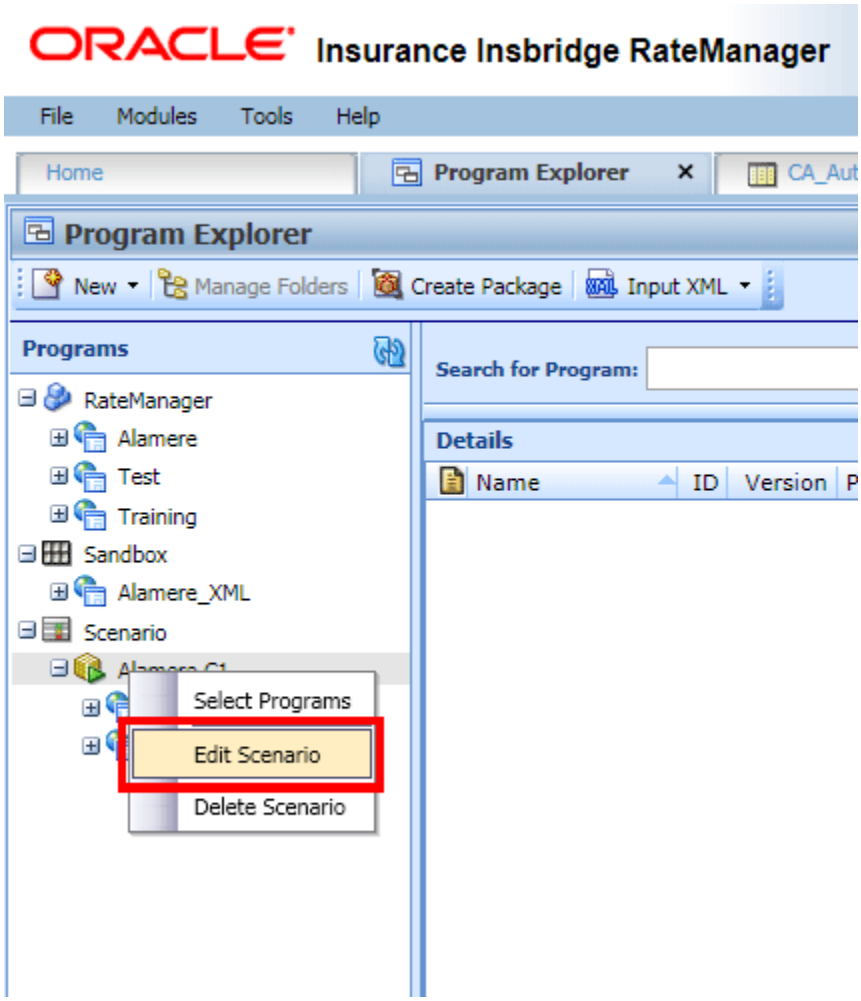


Figure 30 Editing a Scenario

1. Select the scenario where you want to edit the name.
2. Right click and click Edit Scenario.

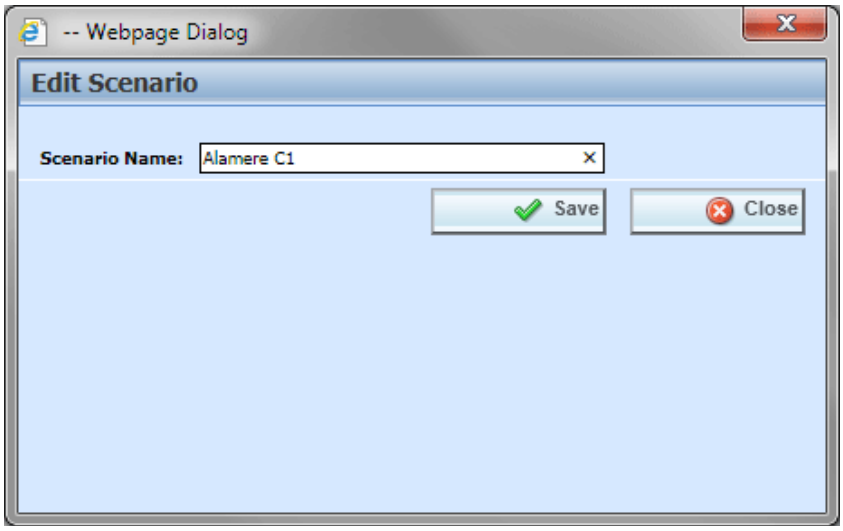


Figure 31 Editing a Scenario Name

3. Edit the name and click Save. The screen closes and the scenario is updated. Click Close to close the screen and cancel editing the scenario.

PUBLISHING A SCENARIO

When table changes/updates are complete, the subversion can be published. Publishing brings the updated subversion program from the Scenario domain back to the RateManager domain.

When the Scenario manager publishes a program, there are two options:

- **Programs can be merged with an existing program**. This option only updates the tables in the original program. No new program version or table revisions are created. Changes made to the subrevision tables in the subversion program will be brought in to the program. The updated program in the RateManager domain type can be edited as needed. To test, you must package the program. Changes are reflected when the program is packaged.

- **Create a new program version in the RateManager domain**. A new program version is created. New table revisions are also created and the active revision is updated to the revision used by the newly created program version. The program in the RateManager domain type can be edited as needed. To test, you must package the program. Changes are reflected when the program is packaged.

Publishing any program in the scenario locks the entire scenario. If the scenario is no longer needed, it can be deleted.

1. In the Scenario domain, select the program you want to publish. Open the program to the Program Details tab.

2. Click the Publish option on the top bar menu.

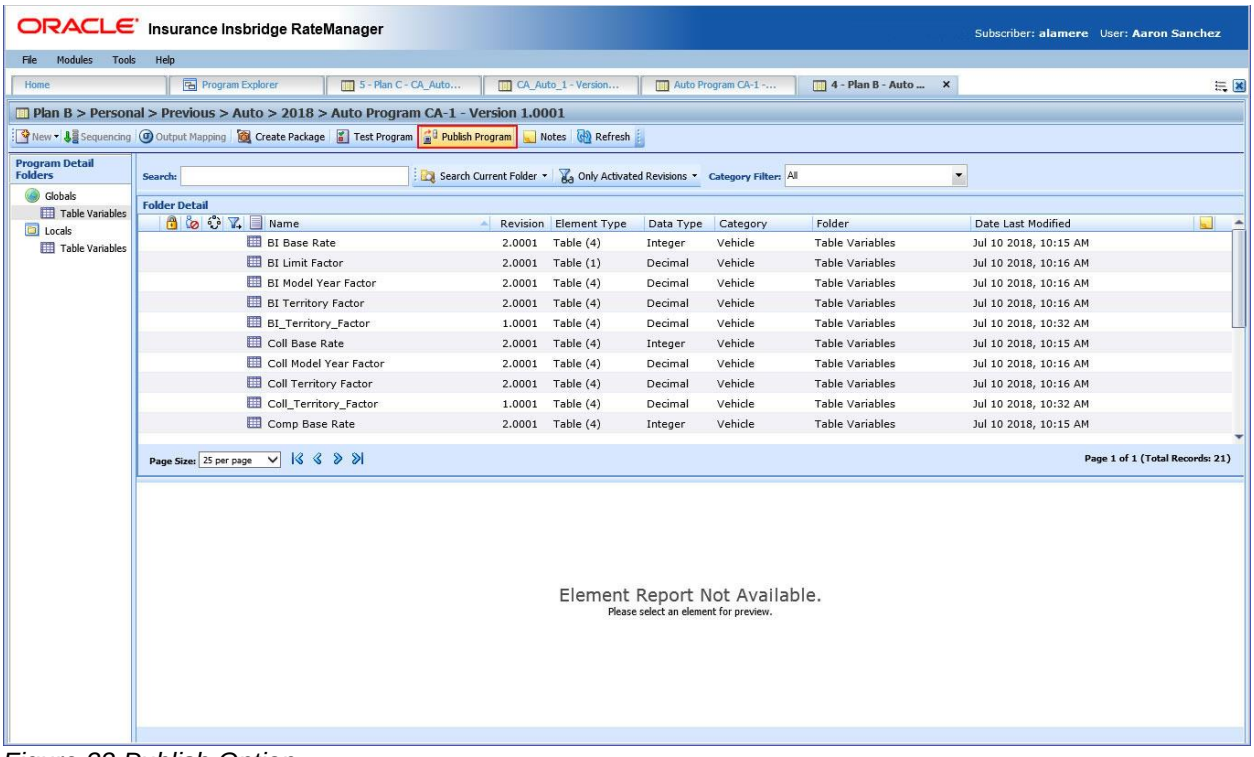


Figure 32 Publish Option

3. The process begins with a program compare report. The baseline program is the RateManager program originally selected to be added to the scenario. The comparison program is the program you are currently in.

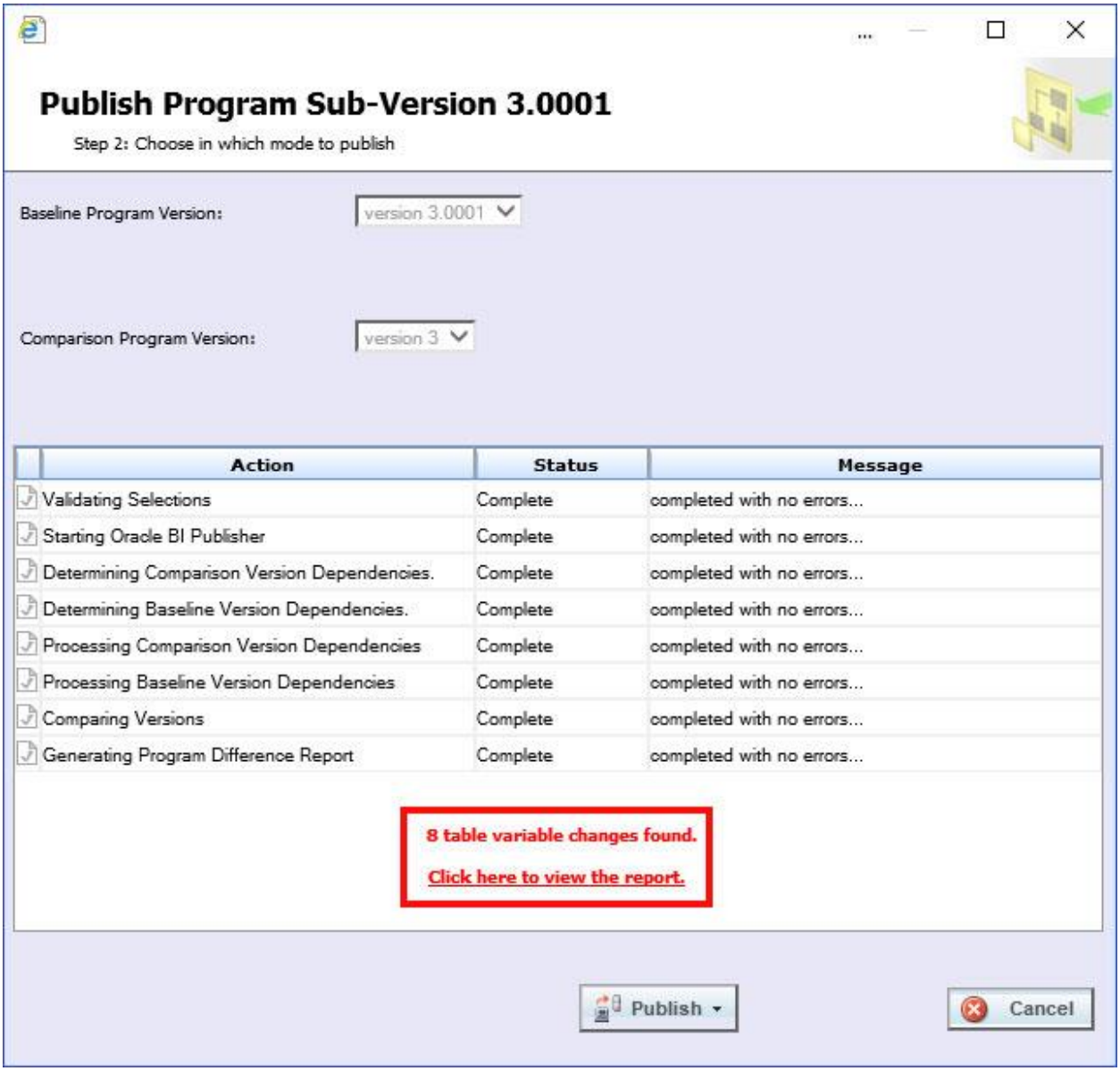


Figure 33 Program Compare Report

4. When the report completes, results are listed. Click to view the report. After viewing the report, you can click Cancel to stop publishing.

5. Select the publish method you need: Merge or Create New.

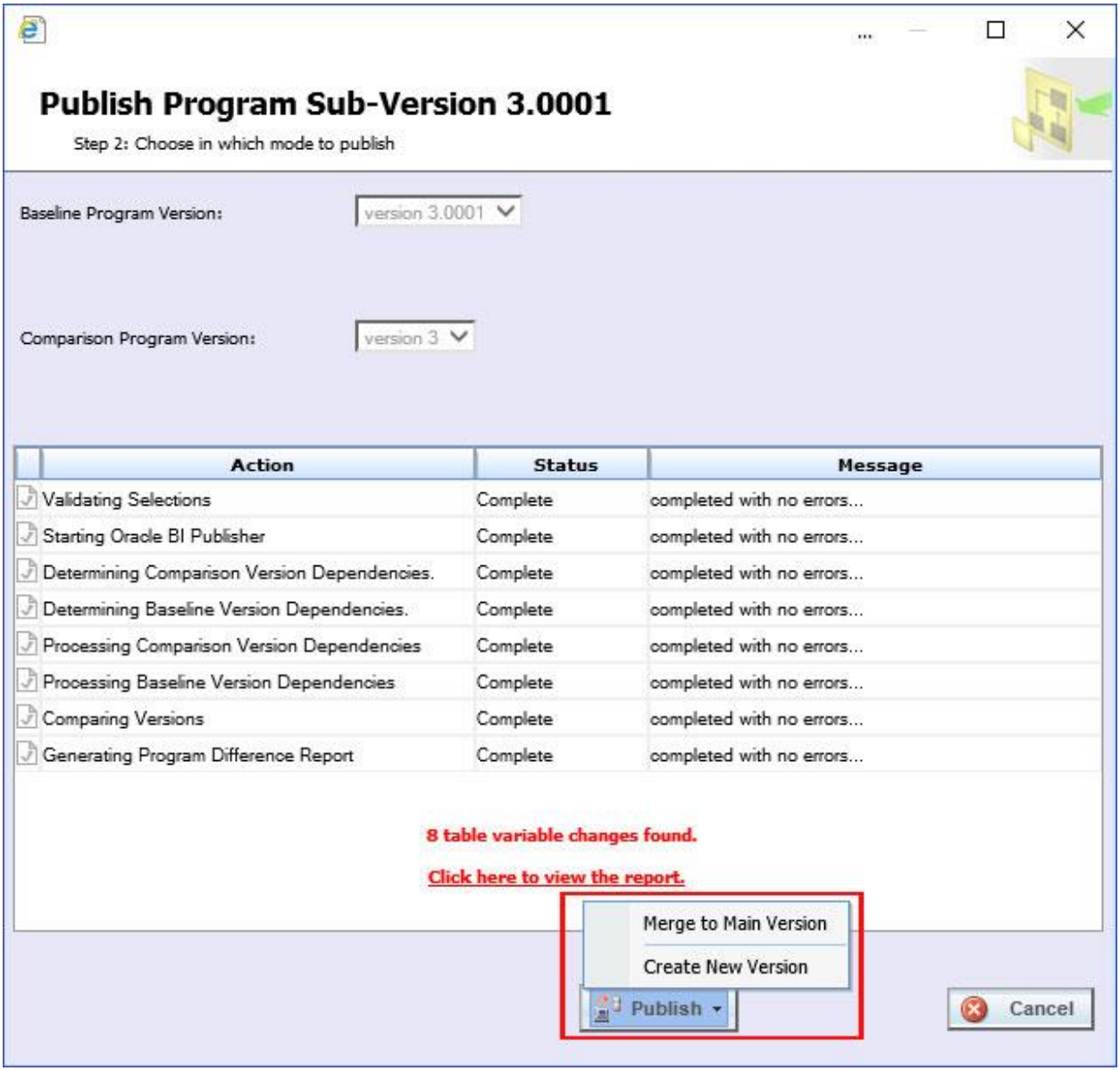


Figure 34 Publishing Options for a Scenario

6. The program will be published how you selected. When publishing is complete, a success or fail message is presented. The scenario is now locked. The program that was published will have a published icon.

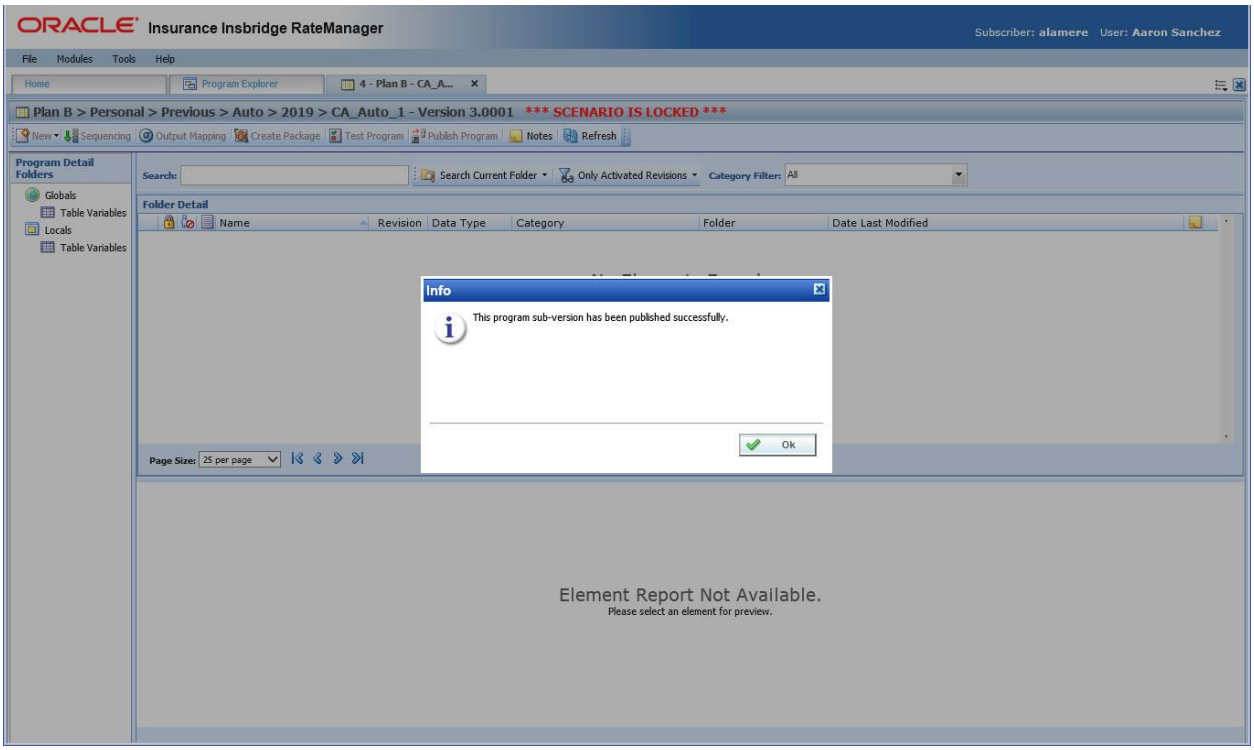


Figure 35 Completed Publish Program

Locked Scenarios

The main purpose of locking the published scenario is to prevent multiple “publish” actions for one scenario. When all changes are done and testing has been completed, the program is published and work in the scenario is done. Publishing should be the final action taken. To prevent any additional changes, the scenario is locked. All testing of table changes should be done in the scenario sandbox before publishing.

A locked scenario prevents multiple change-publish actions but serves as a reminder that a program has been published back to the RateManager domain type.

If you need to test additional changes after publishing, create a new scenario sandbox and add the program back in.

Unlocking a Scenario

If a scenario needs to be unlocked, a system administrator can unlock it. Be aware that unlocking a scenario holds the potential to overwrite the work that has been done to a published program.

1. Select the scenario that you want to unlock.
2. Right click and click Unlock Scenario.
3. A confirmation message is displayed. Click OK to unlock. Click Cancel to stop.

Figure 36 Unlock Confirmation Message

4. The scenario icon will change from locked to open. The scenario is now unlocked.

Deleting a Scenario

You must have Scenario Manager Permissions to create, edit and delete scenarios.

Scenarios can be deleted at anytime. The entire scenario is removed. Deleting a scenario does not delete the programs or tables from the RateManager domain type.

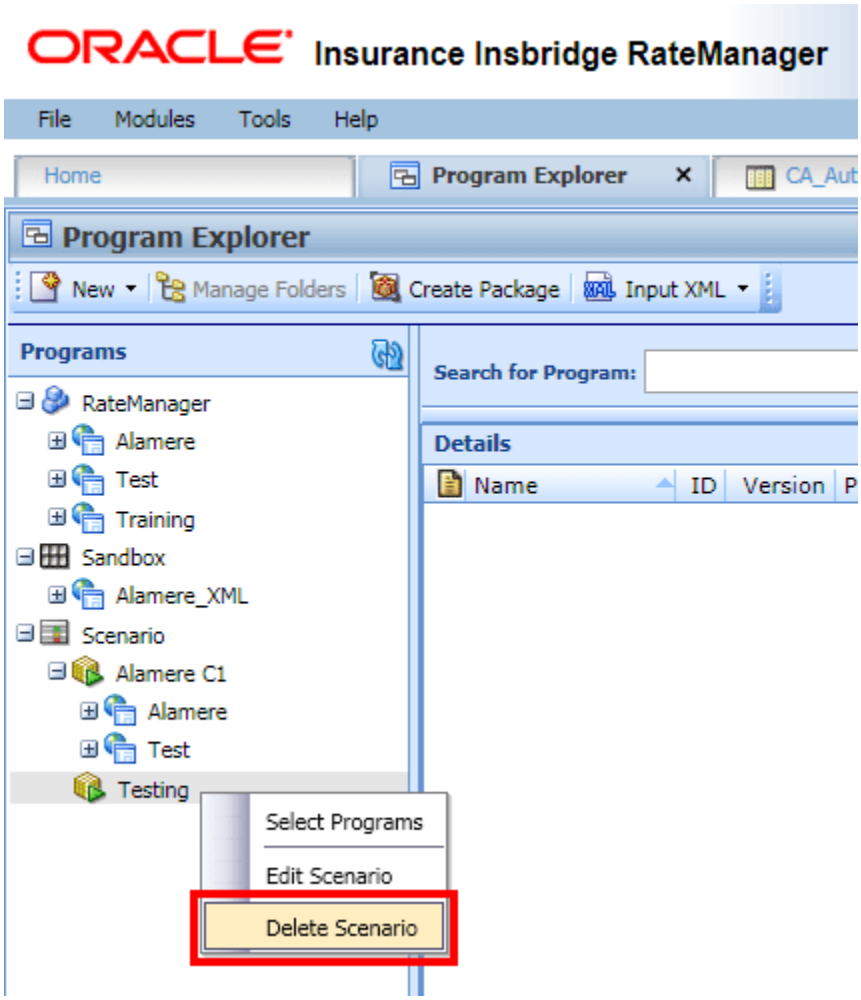


Figure 37 Deleting a Scenario

1. Select the scenario that you want to delete.
2. Right click and click Delete Scenario.

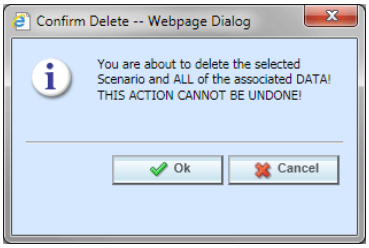


Figure 38 Deleting a Scenario Message

3. A confirmation is displayed. Select OK to delete the scenario. The screen closes and the scenario is deleted. Click Cancel to close the screen and cancel deleting the scenario.

Locked scenarios can be deleted.

SCENARIO USER

Most often the scenario user will not have RateManager domain type access. Only allowing scenario access helps to prevent the other programs and program elements from being unintentionally edited. The only items in Program Explorer will be scenarios. The scenarios contain the products where the user has scenario permissions. The products contain the programs that have been added by the scenario manager. The programs only contain the tables that have been added by the scenario manager.

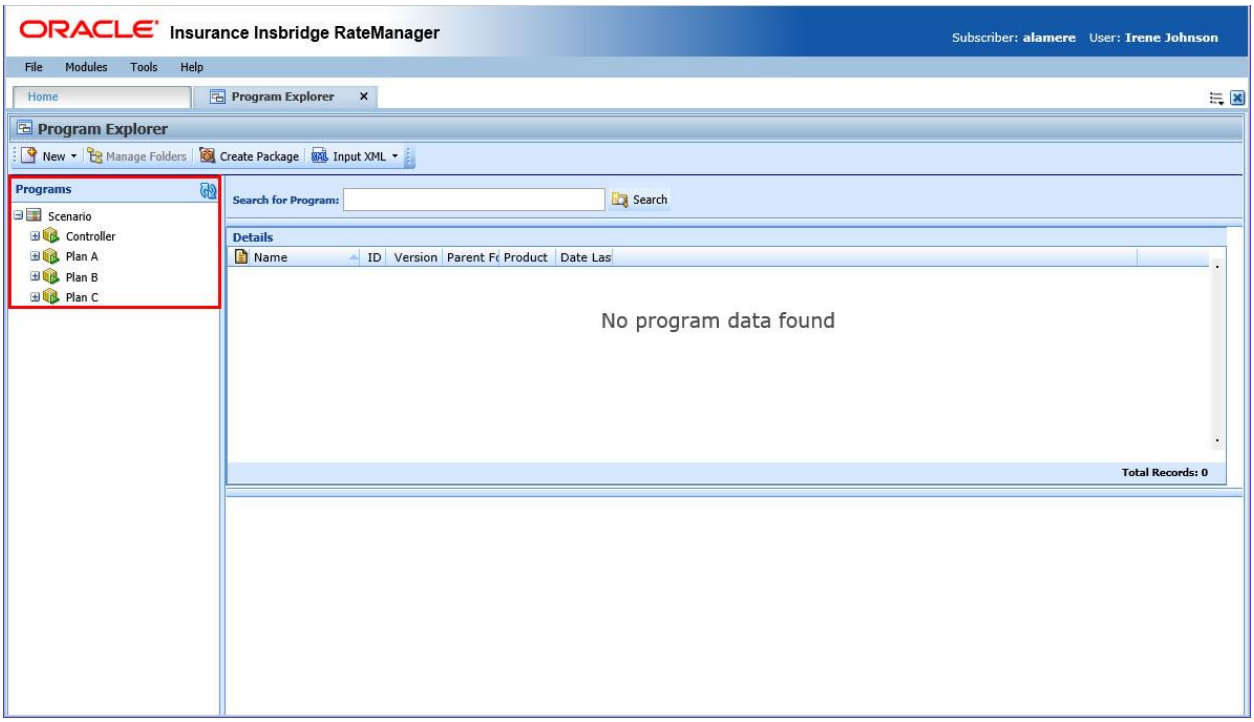


Figure 39 Program Explorer for a Scenario User

A Scenario user can:

- Import or Export Table Data
- Edit Table Data
- Update the Default Value
- Add or remove criteria
- Add Notes to the Table
- View Reports
- View Notes for the Program

Other options such as adding or deleting linked variables or advanced options are not allowed.

Scenario User – Program Details

On Program Details, a scenario user has access to tables only. The tables listed are the tables that have been added by the scenario manager. Right click menus are available with the Copy or Delete functions grayed out and unavailable.

If a table is not listed, please contact your scenario manager or system administrator.

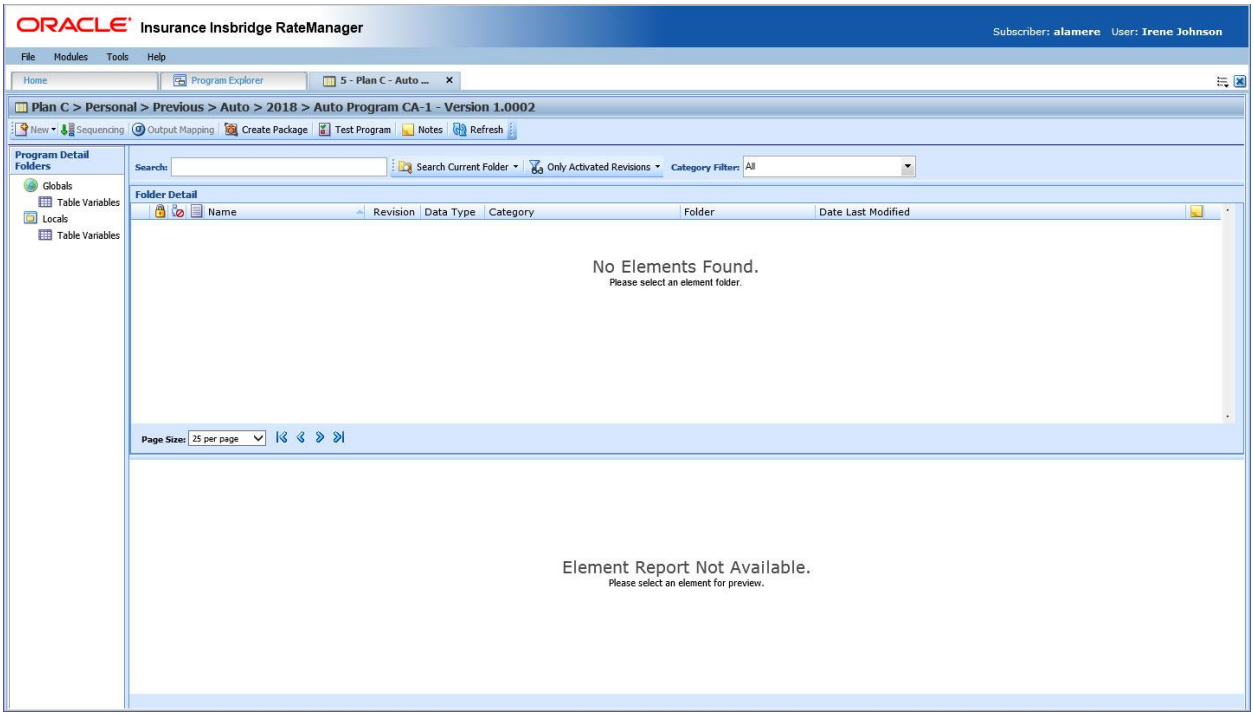


Figure 40 Program Details for a Scenario User

Some options on the Program Details tab are grayed out and unavailable, for example, sequencing and output mapping.

After editing, programs can be packaged and tested. Permissions are required to package.

Scenario User – Editing a Table

Scenario users can edit tables including importing/exporting data, adding or removing criteria, and setting a default value. Some options are grayed out and unavailable, for example, adding a linked variable or advanced options.

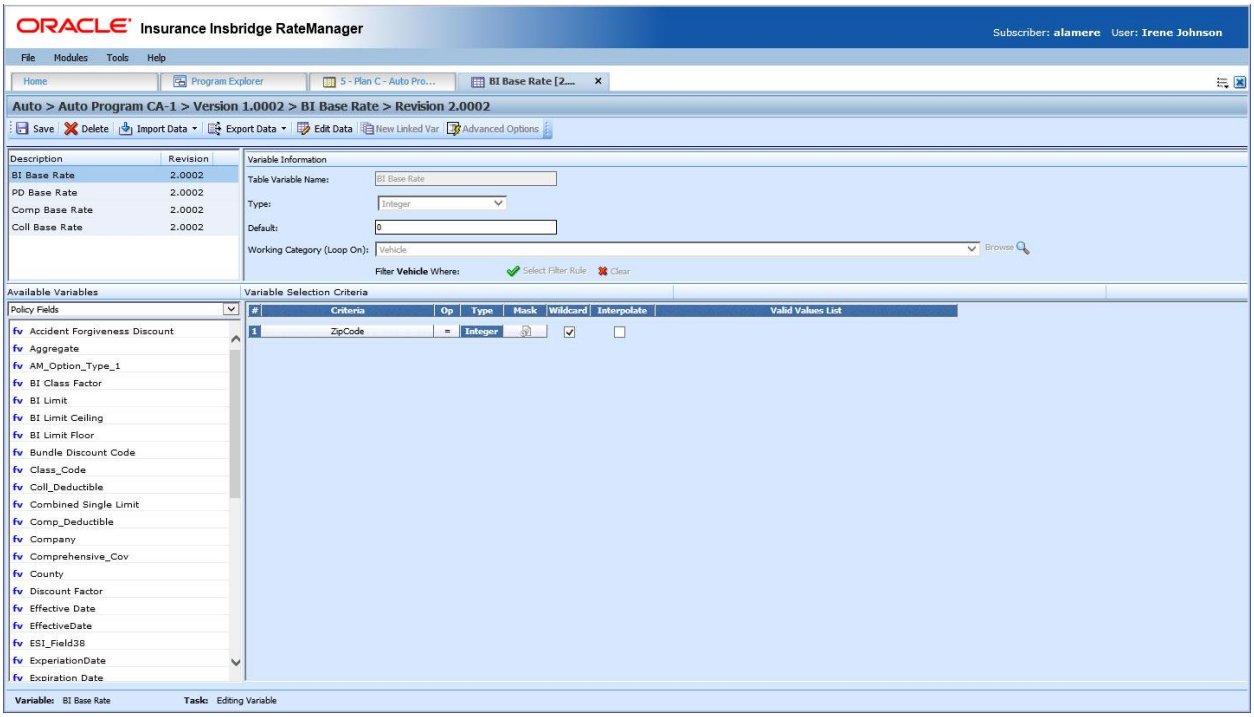


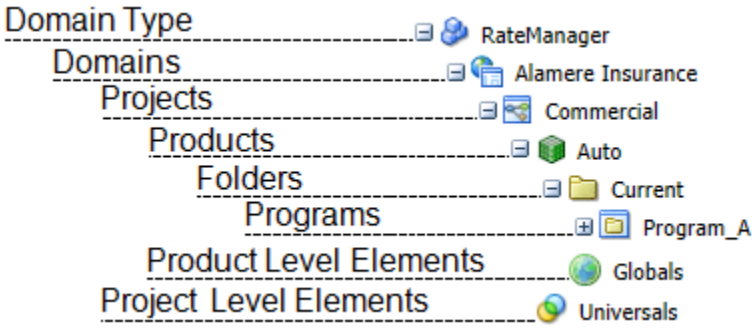
Figure 41 Editing a Table for a Scenario User

If an option is not available, for example, an “access denied” message is displayed when you try to import or export a file, please contact your system administrator. Scenario users cannot create new field or result variables.

If universal permissions have been granted, tables will be available for editing on the universal tab for the product.

DOMAIN TYPES AND DOMAINS

RateManager allows for the structure and classification of data and relationships to be expansive. This allows you to create programs under a wider scope where categories, fields and other elements can be shared across a wider array of programs and lets you create programs that are still line of business specific but share basic information with other programs in separate lines of business.



DOMAIN TYPES

A domain type is a top level is container that defines the data source. There are three domain types:

- **RateManager** – the default domain type utilizing the RateManager database. This domain type cannot be edited or deleted.

- **Other** – a user-created domain type where outside data sources brought in to RateManager can be applied. For example, library templates can only be applied to an Other domain type. Other domain types also can be used to create and manage programs like the RateManager domain type. Other domain types are open for editing and can be deleted if no longer needed.

- **Scenario** – a domain type designed for users who only need to edit table data. Within a Scenario domain type are user created scenarios where tables can be opened within a program and made available for editing by users with scenario user permissions.

In most instances, the RateManager domain type is listed on the Program Explorer page.

RateManager and Other domain types are visible to all users. Scenario domain types are only visible to users with scenario permissions. You must have Project permissions to manage domain types. If the domain options are grayed out or ineffective, then you do not have the permissions to work in these areas. Contact your system administrator to obtain permissions.

Other User Created Domain Type

The Other domain type is a user created domain type that must be used for template and import programs as well as public XML. Other domain types also can be used to create programs from scratch. The functionality is the same as RateManagers.

There can only be one RateManager domain type. You can have multiple Other domain types. Each Other domain type requires its own setup and is independent of any other domain type. No data is shared between them.

For example, you can create an Other domain type for each template applied. You also may be required to create a new Other domain type if there are category or field differences between existing templates and the new template you want to apply.

Other domain types can be created, edited and deleted.

Scenario Domain Type

The Scenario domain type is visible to users with scenario permissions. For more on scenarios, please see Chapter 4 Scenario.

New Other Domain Type

Other domain types can be created at any time.

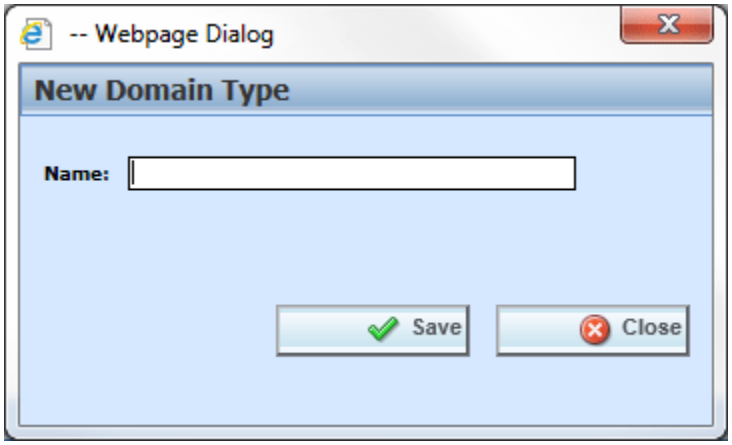


Figure 42 Creating a New Other Domain Type

1. On the Program Explorer tab, select the New option in the top bar menu.
2. Select Domain Type.
3. Enter in the name and click Save. Domain type names can be 50 alpha numeric characters with selected special characters.

NOTE: *& * < > ? are not allowed.*

4. The screen refreshes and the user created domain type is listed on Program Explorer.

Once the domain type has been established, domains can be added.

Editing an Other Domain Type

On an Other domain type, only the name can be edited. The RateManager domain type does not allow for editing.

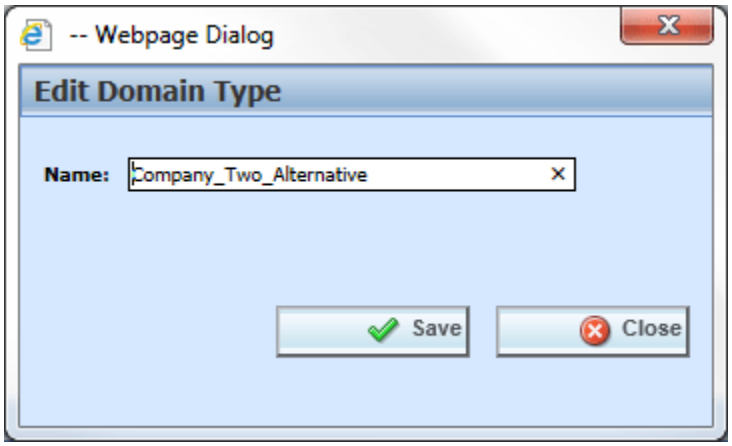


Figure 43 Editing a Domain Type

1. Select the Other domain type where you want to change the name.
2. Right click and select Edit Domain Type.
3. Make your changes and click Save.

Deleting an Other Domain Type

Deleting a domain type removes the entire domain type from the system. Everything in the domain type, including projects, products, folders, programs, and data is deleted.

Deleting a domain type should be done only when nothing in the domain type is needed. Make sure this is the action you want to take. A deleted domain type cannot be restored within RateManager. If a domain type has been deleted by mistake, please contact Oracle Support.

When deleting a domain type, you are asked to confirm your selection twice before the domain type is deleted. The RateManager domain type cannot be deleted. Any Other user created domain type or PC domain type can be deleted at any time.

You must have project permissions to delete a domain type. If the delete domain type option is grayed out or unresponsive, you may not have permissions.

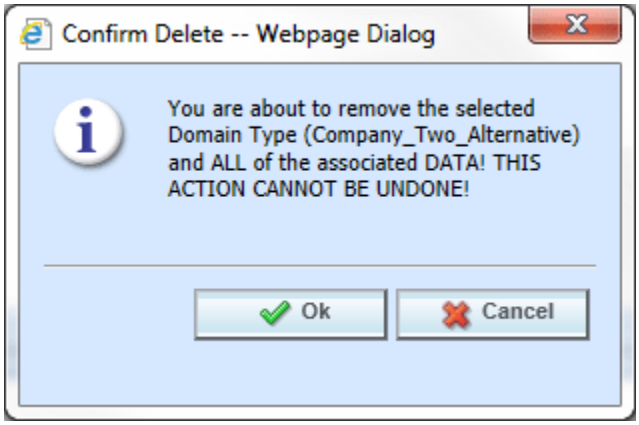


Figure 44 Deleting a Domain Type

1. Select the Other domain type that you want to delete.
2. Right click and select Delete Domain Type.
3. You will be asked to confirm the deletion. Click OK to continue. Click Cancel to stop the process and close the screen.
4. The confirmation screen is presented. You will be asked to confirm the deletion again, Click OK to delete the domain type. The screen refreshed and the domain type is removed.

Other Domain Type Right Click Menu Options

The RateManager domain type has one right click option, New Domain. Other domain types have three options.

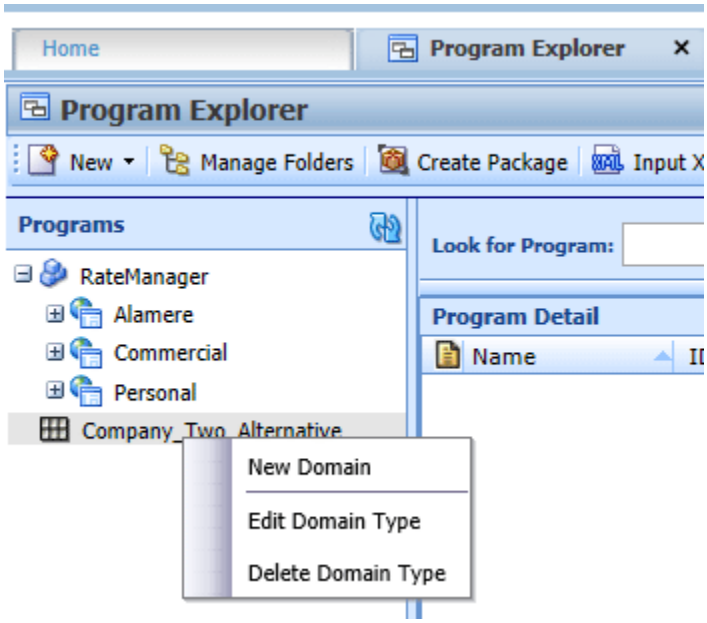


Figure 45 Other Domain Type Right Click menu

- **New Domain** – Creates a new domain under the selected domain type.
- **Edit Domain Type** – Opens the edit screen for the selected domain type.
- **Delete Domain Type** – Deletes the selected domain type.

If any of these options are grayed out or unresponsive, you may not have the proper permissions. Please contact your system administrator.

DOMAINS

Domains are classifications that fall in between domain types and projects that can serve as high level containers where projects can be grouped. At least one domain is required for projects but more can be added if needed. Domain names cannot be duplicated across domain types.

New Domains

New domains can be created at any time.

1. Select the domain type where you want to create a new domain.

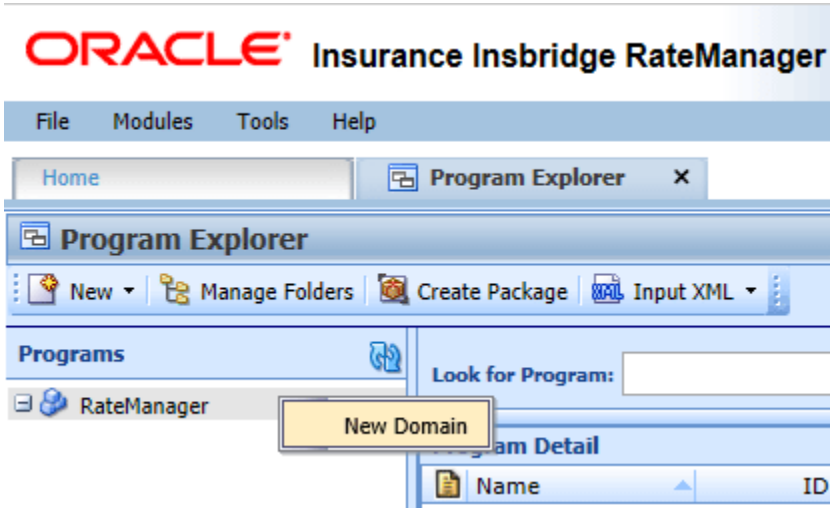


Figure 46 Right Click for New Domain

2. Right click and select New Domain. The New Domain screen is shown.

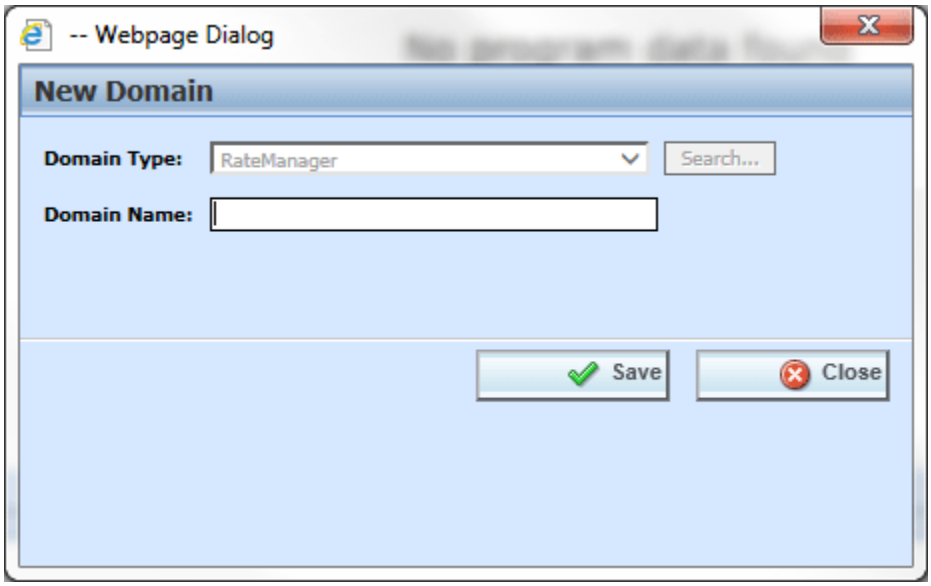


Figure 47 New Domain

3. In PC domain types, you can select the domain type source. This is only available in PC domain types. For RM or other domain types, this option is not available.

4. Enter in a Name. Names can be up to 50 alpha numeric characters.

5. Click Save to save your entry. After the Program Explorer tab refreshes, the new domain is listed.

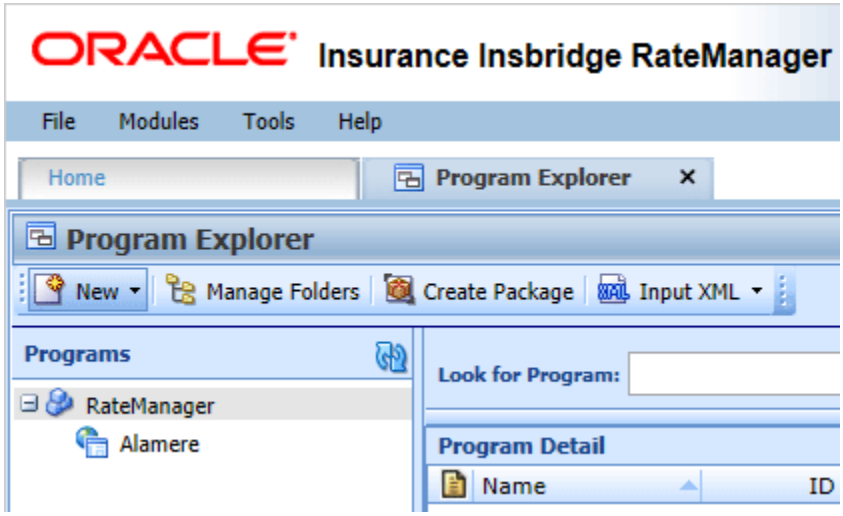


Figure 48 Program Explorer with New Domain

After domains have been added, projects can be added.

Editing Domains

Editing allows for the name to be changed.

1. Select the domain type and domain where you want to edit.

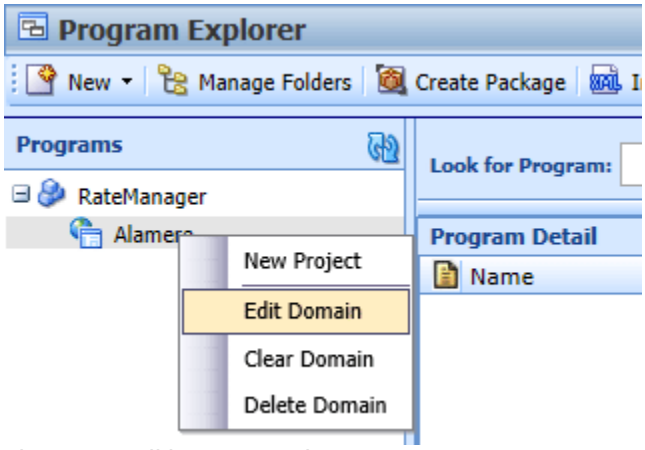


Figure 49 Editing a Domain

2. Right click and select Edit Domain. The Edit Domain screen is shown.

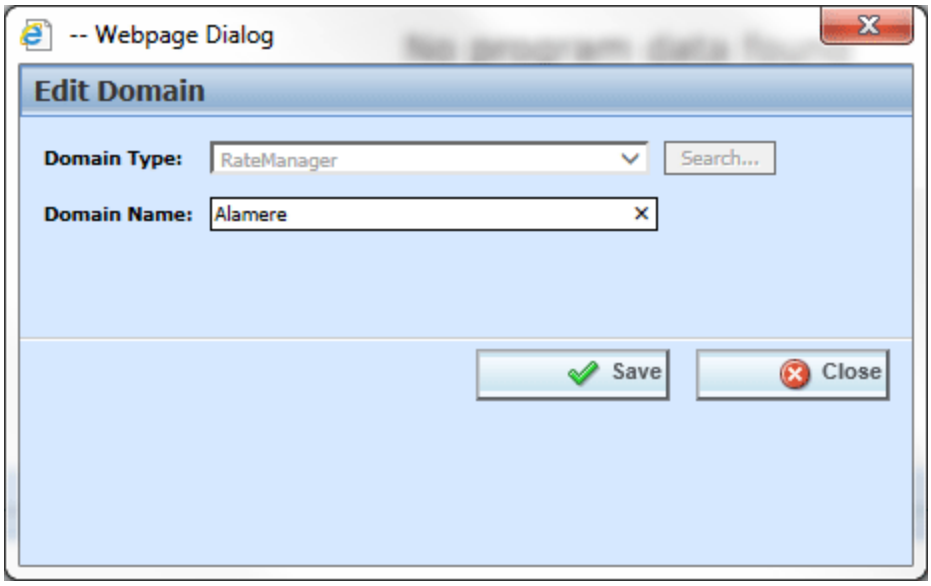


Figure 50 Editing a Domain on the Edit Domain Screen

3. Make your changes.
4. Click Save. After the Program Explorer tab refreshes, the changes are shown.

Clearing Domains

Clearing a domain removes all content from the domain. All projects, folders, programs, and data, including categories, fields, variables, sequences, and mappings, are cleared. Clearing an Other user created domain allows you to create, or import content into a clean environment. For example, if you try to import a program and receive a notice that the category structure does not match, you will not be able to import that program in to the domain. You can create a new domain or you can clear the old domain if you no longer need any content in there. Clearing the domain removes the previous category structure and enables you to apply the new import program.

Clearing a domain should be done only when nothing in the domain is needed. Make sure this is the action you want to take. Once a domain is cleared, there is no undo. You are asked to confirm your selection twice before the domain is cleared. Any domain can be cleared at any time.

You must have domain management permissions to clear a domain. If the clear domain option is grayed out or unresponsive, you do not have permissions.

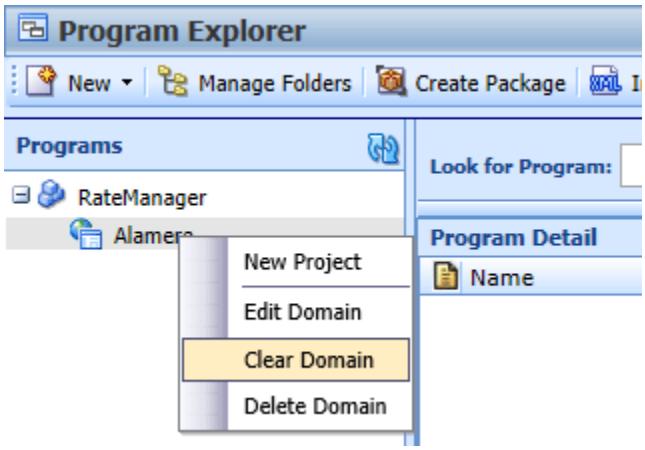


Figure 51 Clearing a Domain

1. Select the domain type and the domain that you want to clear.
2. Right click and select Clear Domain.

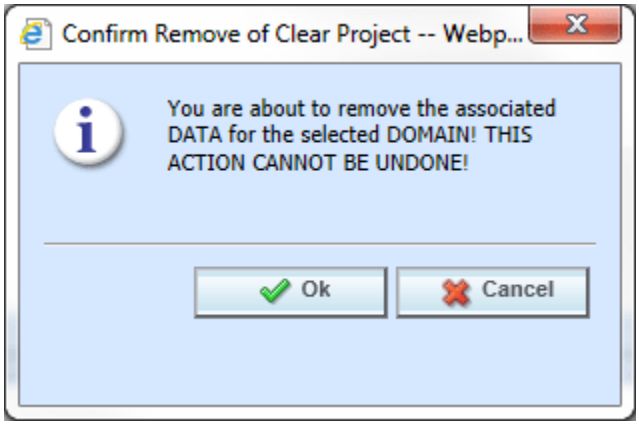


Figure 52 First Confirmation to Clear the Domain

3. Click OK to clear the domain.

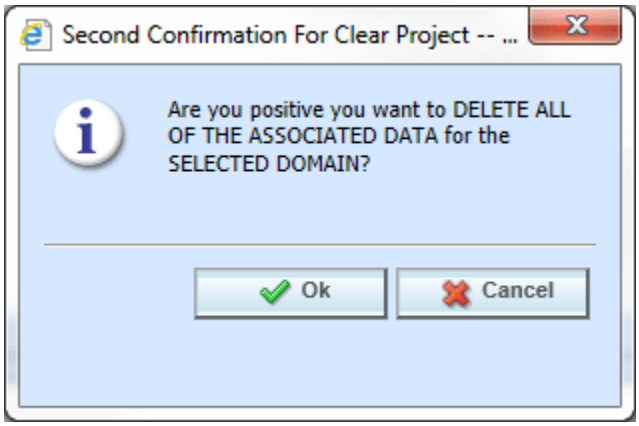


Figure 53 Second Confirmation to Clear the Domain

4. A second confirmation screen is shown. Click OK to clear the domain. After the Program Explorer tab refreshes, the changes are shown. The domain type will remain. Select **Cancel** to return to the Program Explorer tab without clearing the domain.

Deleting Domains

Deleting a domain removes the domain from the system. Everything in the domain, including projects, folders, programs, and data is deleted.

Deleting a domain should be done only when nothing in the domain is needed. Make sure this is the action you want to take. Once a domain is deleted, there is no undo. You are asked to confirm your selection twice before the domain is deleted. Any domain can be deleted at any time.

You must have domain management permissions to delete a domain. If the delete domain option is grayed out or unresponsive, you may not have permissions.

1. Select the domain type and domain that you want to delete.

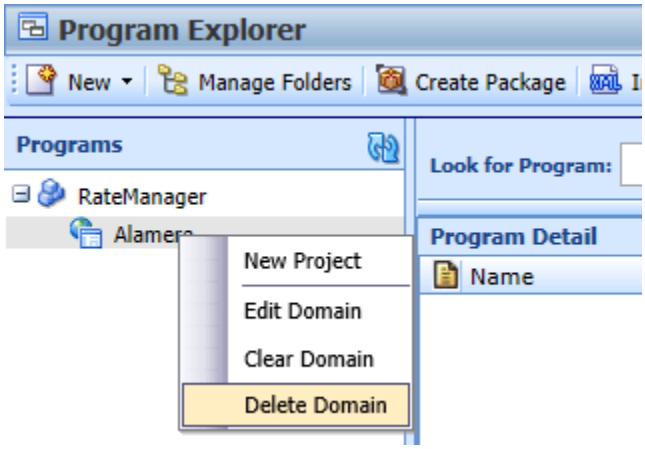


Figure 54 Deleting a Domain

2. Right click and select Delete Domain.

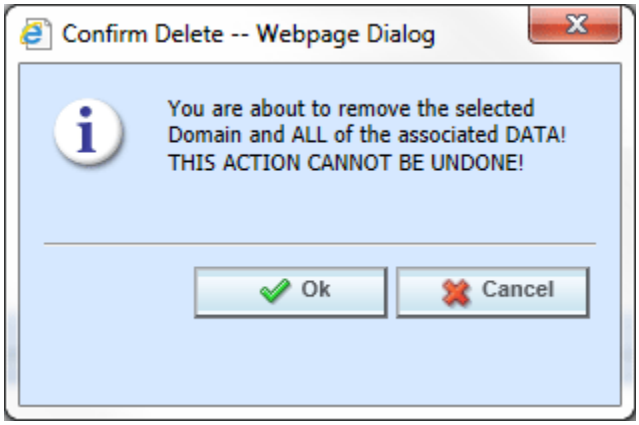


Figure 55 First Confirmation to Delete the Domain

3. Click OK to delete the domain.

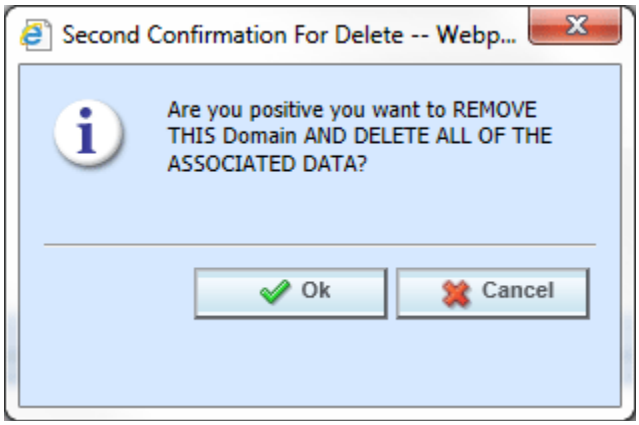


Figure 56 Second Confirmation to Delete the Domain

4. A second confirmation screen is shown. Click **OK** to delete the domain. After the Program Explorer tab refreshes, the changes are shown. Select **Cancel** to return to the Program Explorer tab without deleting the domain.

Domain Right Click Menu Options

All domains have the same right click menu options.

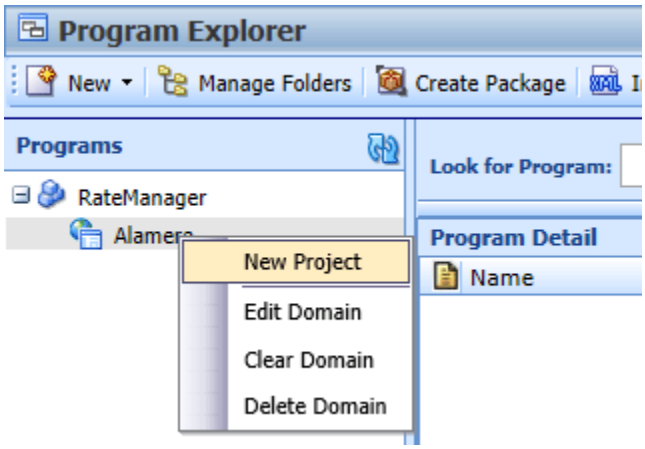


Figure 57 Domain Right Click menu

If any of these options are grayed out or unresponsive, you may not have the proper permissions. Please contact your system administrator.

PROJECTS AND PRODUCTS

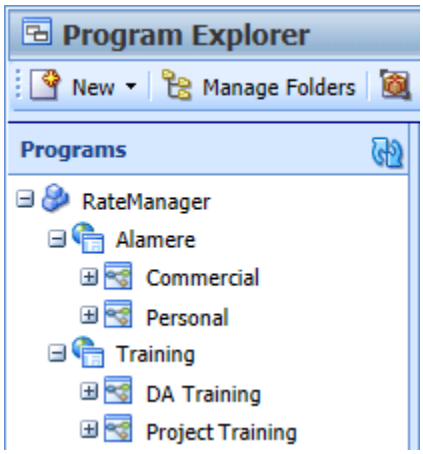
Projects can hold one or more products. This level also holds universal categories and elements that can be used by any program in the project. Products can hold one or more folders. This level also holds global elements that can be used by any program in the project.

If projects or products are not set up prior to working on programs, please contact your system administrator.

NOTE: *Project management is available to system administrators and users with manage projects permissions. Product management is available to system administrators and users with manage products permissions.*

PROJECTS

Projects are classifications that fall in between domains and products. Projects allow for a more precise grouping of products or for a broad grouping of products. Universal elements are at the project level and are unique to the project. Universal elements are not shared across projects.



Projects are required and must be in place prior to products being added. Projects permissions allow you to manage projects in any domain or domain type.

Domains can hold one or more projects. Projects structure and content is determined by the user. RateManager does not place restrictions on what products or how many products can be included in a project.

When creating a new project, the option to include driver assignment is presented. This option is for including the Auto product and driver assignment variables in the project. Including driver assignment will allow the Auto product to be selected and will populate the global and local program detail screen with the driver assignment variable. If you do not select driver assignment, the Auto product is not available. You must select this at the creation of the project. It cannot be added later.

This does not add driver assignment to other products. Only the Auto product can have a driver assignment.

New Projects

New projects can be created at any time. Projects are separate entities and universal information is shared within that project only. Creating a new project is starting from scratch. Everything will be new and unique to that project.

1. Select the domain where you want to create a project.

2. Right click and select New Project.

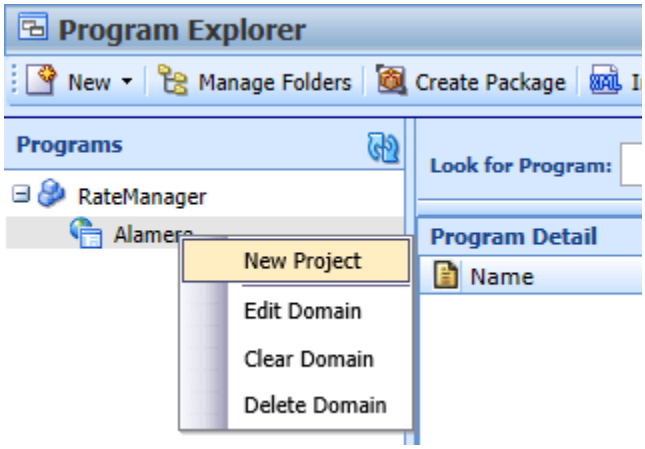


Figure 58 Domain Right Click Menu for New Project

3. The domain type selection is for reference only. No selection can be made.

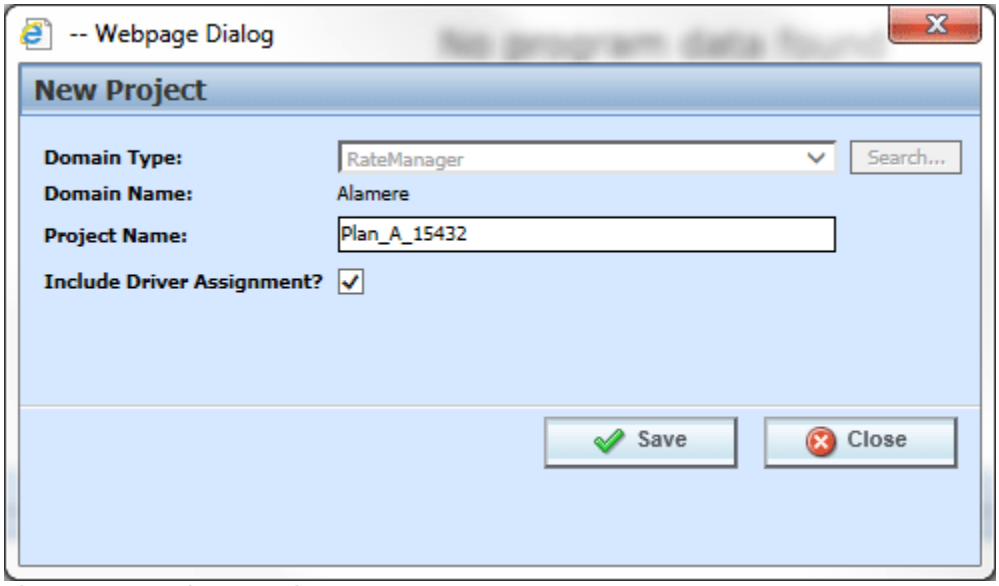


Figure 59 Creating a Project

4. Enter in a project name. Project names can be 50 alpha numeric characters with selected special characters.

5. Check if driver assignment is included. This allows for the Auto product to be selected in the project. This option cannot be added later.

6. Click Save.

Once a project has been created, products can be added from the Select Product screen.

Editing Projects/Selecting Products

Editing a project and selecting the products to be included is done on the Select Product screen.

Product rights are required to work on the Select Products screen. If the options are grayed out or you receive a message that you do not have permissions, please contact your system administrator.

P2P Abbreviation Code

A P2P Abbreviation Code is an optional code that can be added to a project. This feature allows you to tag a project with a code to be used when working with cross project callouts. When you create a cross project relationship, this code will be added to result variables so you can tell what project the result came from as well as what product. This abbreviation is necessary as the same product can be used in multiple projects.

A P2P Abbreviation can be 5 alpha numeric characters.

Selecting Products

Products must be selected in a project. Products are line of business classifications. All activated or deactivated products are listed on the Select Products screen in alphabetical order. You must select the products you want to include. If a product is not selected, it will not be displayed under the project.

If the product you want is not listed on the Select Product screen, it may be disabled on the Manage Products screen. If you have product permissions, you can access the Manage Products screen from the Tools menu. If Manage Products is not available to you, please contact your system administrator to verify that the product has not been disabled.

A product may be listed on the Select Products screen but may not have permissions assigned to it. Group permissions must be assigned to products in order for users to work in them. If you see an Access Denied to all Products message, you may not have permissions to work on the product.

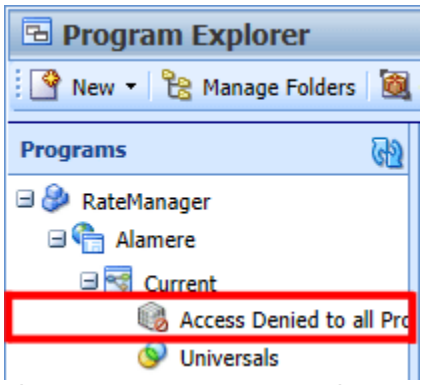


Figure 60 No Access to Products

If you select multiple products and one product is not displayed, you may not have permissions to work on that product. Verify the product has been included in the project. Open the Select Products screen. If

there's a check in front of the product, it has been included. Only the products you have permissions to access are displayed in Program Explorer.

If any product is not available to you, please contact your system administrator to ask about permissions.

Products can be added to a single project one time. After a product is selected in a project, it cannot be selected again. However, products can be repeated across multiple projects. Products that are a repeated in multiple projects are unique to the project where they are located. The actual content of the product is not repeated.

For example, the Umbrella product was selected in both the commercial and personal projects. Any content entered in one will not be automatically entered in the other. They are independent of each other.

Projects can hold one or more products. Project/Product structure and content is determined by the user. RateManager does not place restrictions on what products or how many products can be included in a project.

Editing a Project

Editing a project is done on the Select Product screen. Projects can be edited to change the name and P2P Abbreviation and to add or remove products. Driver assignment selection cannot be changed.

1. Select the project where you want to edit.
2. Right click and select Select Products.

Figure 61 Selecting Products Right Click Menu

3. The name, P2P Abbreviation and available products are all open for editing.

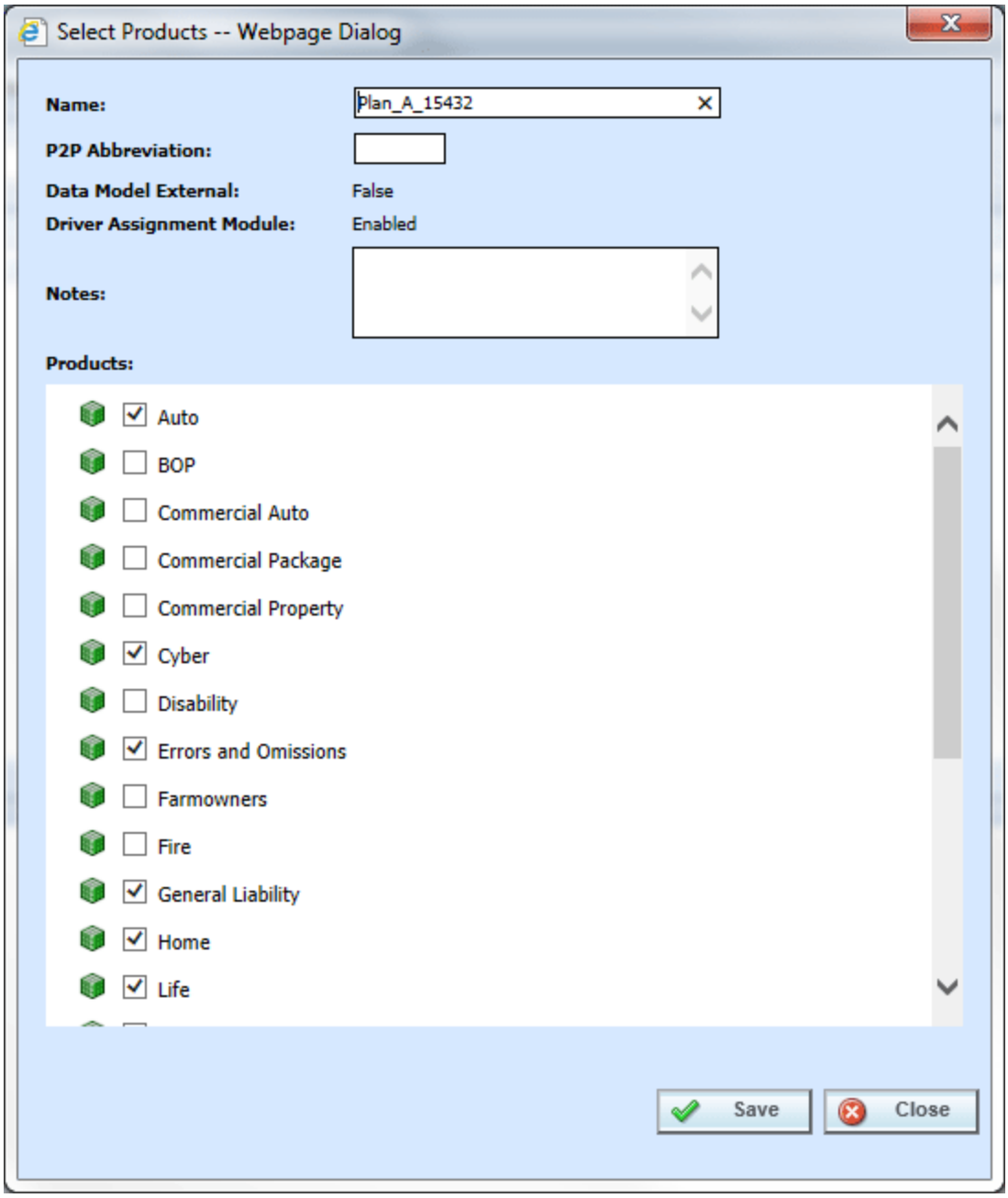


Figure 62 Editing a Project

4. Make any changes including adding notes.
5. Click Save.

Clearing Projects

Clearing a project removes all content from the project. All folders, programs, and data, including categories, fields, variables, sequences, and mappings, are cleared.

Clearing a project should be done only when nothing in the project is needed. Make sure this is the action you want to take. Once a project is cleared, it cannot be restored from within RateManager. You must confirm your selection twice before the project is cleared. Any project can be cleared at any time.

You must have project management permissions to clear a project. If the clear project option is grayed out or unresponsive, you do not have permissions.

1. Select the project that you want to clear.
2. Right click and select Clear Project.

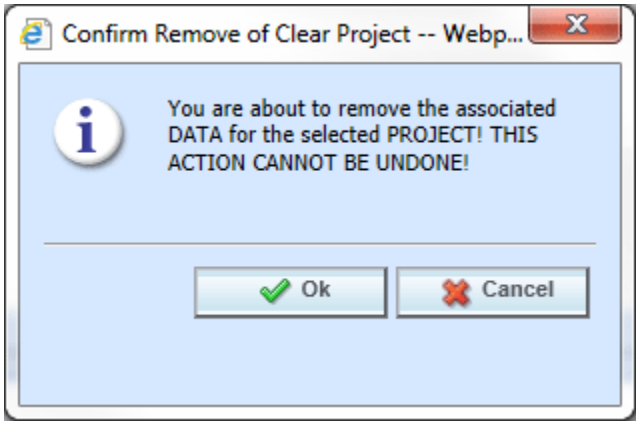


Figure 63 First Confirmation to Clear the Project

3. Click OK to clear the project.
4. A second confirmation screen is shown. Click OK to clear the project. After the Program Explorer tab refreshes, the changes are shown. Select **Cancel** to return to the Program Explorer tab without clearing the project.

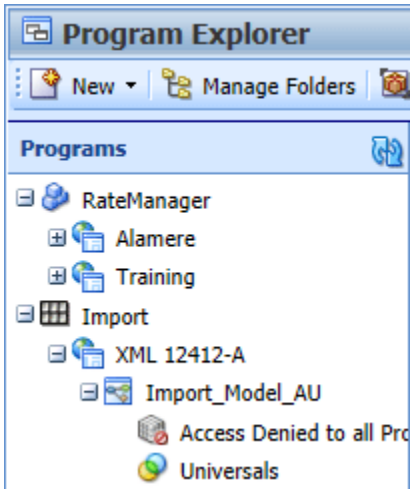


Figure 64 Cleared Project

When a project is cleared, the project remains without any content. Products must be selected again. The Access Denied to all Products message is displayed for cleared projects.

Moving Projects

Moving a project moves the entire project, including products, folders and universal elements, to another domain within the same domain type. The available domains listed are in the same domain type as the project being moved. You cannot create a new domain type and move a project in to it. At the time of the move, the project name can be changed. The products selected in the project and the P2P abbreviation will remain the same but can be edited after the move.

You must have project management permissions to move a project. If the move project option is grayed out or unresponsive, you do not have permissions.

1. Select the project you want to move and right click.

Figure 65 Moving a Project

2. Select Move Project.

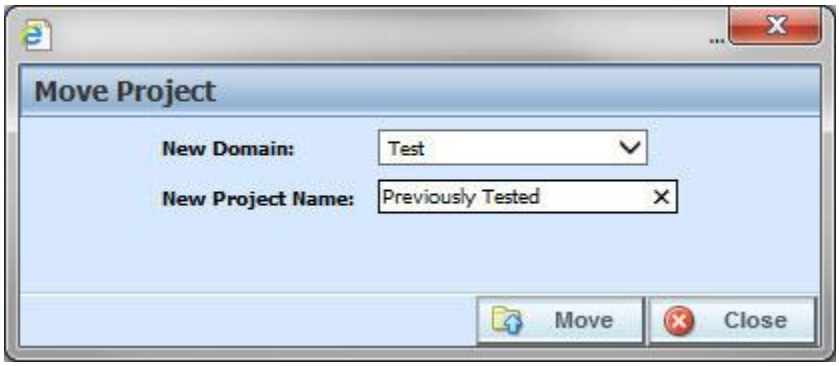


Figure 66 Move Project Options

3. Select the domain where you want to move the project to.
4. The name is open for editing. Other items in the project can be edited after the move.
5. Click OK. The entire project is moved.

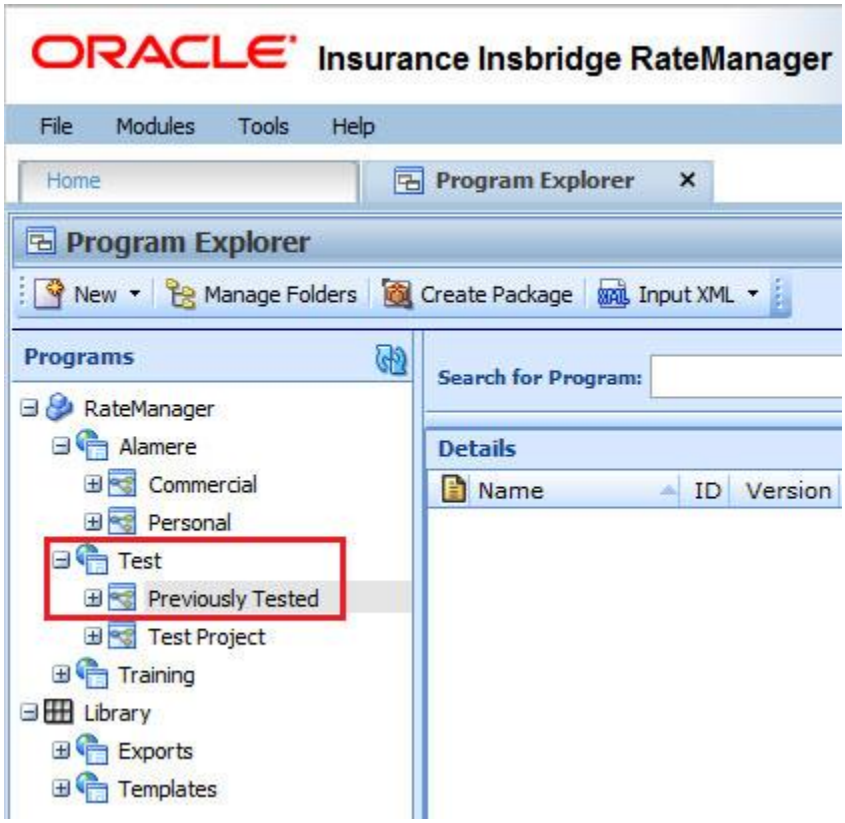


Figure 67 Final Location for Moved Project

Deleting Projects

Deleting a project removes the entire project from the system. Everything in the project, including projects, folders, programs, and data is deleted.

Deleting a project should be done only when nothing in the project is needed. Make sure this is the action you want to take. A deleted project cannot be restored from within RateManager. If a project has been deleted by accident, please contact Oracle Support.

When deleting a project, you are asked to confirm your selection twice before the project is deleted.

You must have project permissions to delete a project. If the delete project option is grayed out or unresponsive, you may not have permissions.

1. Select the project that you want to delete.
2. Right click and select Delete Project.

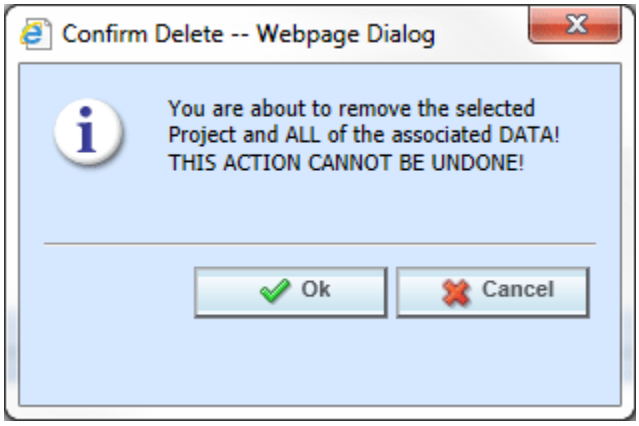


Figure 68 Deleting a Project

3. You will be asked to confirm the deletion. Click OK to continue. Click Cancel to stop the process and close the screen.
4. The confirmation screen is presented. You will be asked to confirm the deletion again, Click OK to delete the project. The screen refreshed and the project is removed.

Project Right Click Menus

All projects have the same right click menu options:

- **Map Custom XML** – Opens the Custom XML Mapping screen where you can create a custom XML mapping for the project.
- **Select Products** – Opens the select products screen for project editing and selecting products for inclusion in the project.
- **Move Project** – Moves the entire project to another domain.
- **Clear Project** – Clears the project of all content.

- **Delete Project** – Deletes the project from the domain.

If any of these options are grayed out or unresponsive, you may not have the proper permissions. Please contact your system administrator.

PRODUCTS

Products are line of business classifications that fall under projects. Products are the holders for one or more folders. This level also holds global elements that can be used by any program in the project. Globals are product specific and are not shared across projects.

When you click a product in the Programs area of Program Explorer, you will see the folders that were created under the selected product in the Program Detail area. To view the programs for that folder, double click the folder. You also can expand the product contents in the Programs area to see all folders and programs under the product.

Products cannot be edited or updated from Program Explorer. Products are managed in the Manage Products option of the Tools menu. The Manage Projects option is available to system administrators and users with manage projects permissions. If projects are not set up prior to working in Program Explorer, please contact your system administrator.

Global Versioning Report

The Global versioning report provides details of the program version criteria and program version data in a report. You can generate the global version report at three levels: the program version level, the program level, and the product level.

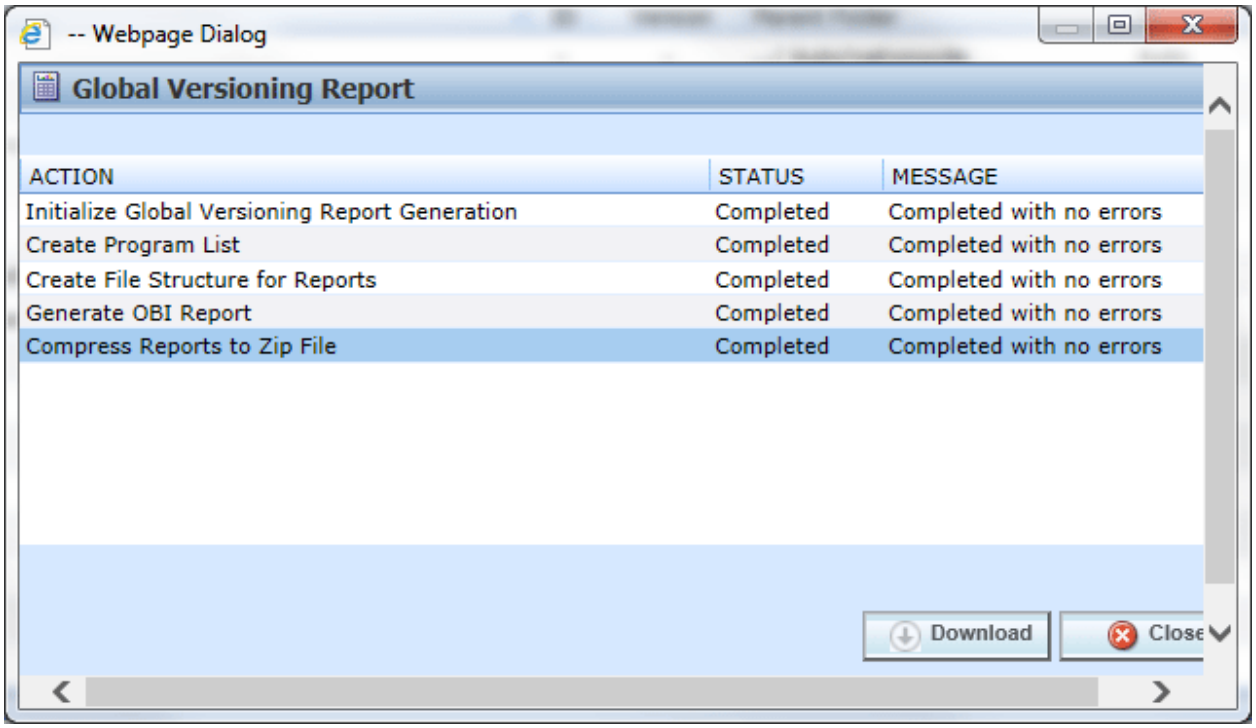


Figure 69 Global Versioning Report Notification

When the global versioning report is selected form the right click menu, a PDF is generated. The Global Versioning Report at the Product level generates a separate PDF report for each program in that product.

Adobe Acrobat Reader is required to view reports.

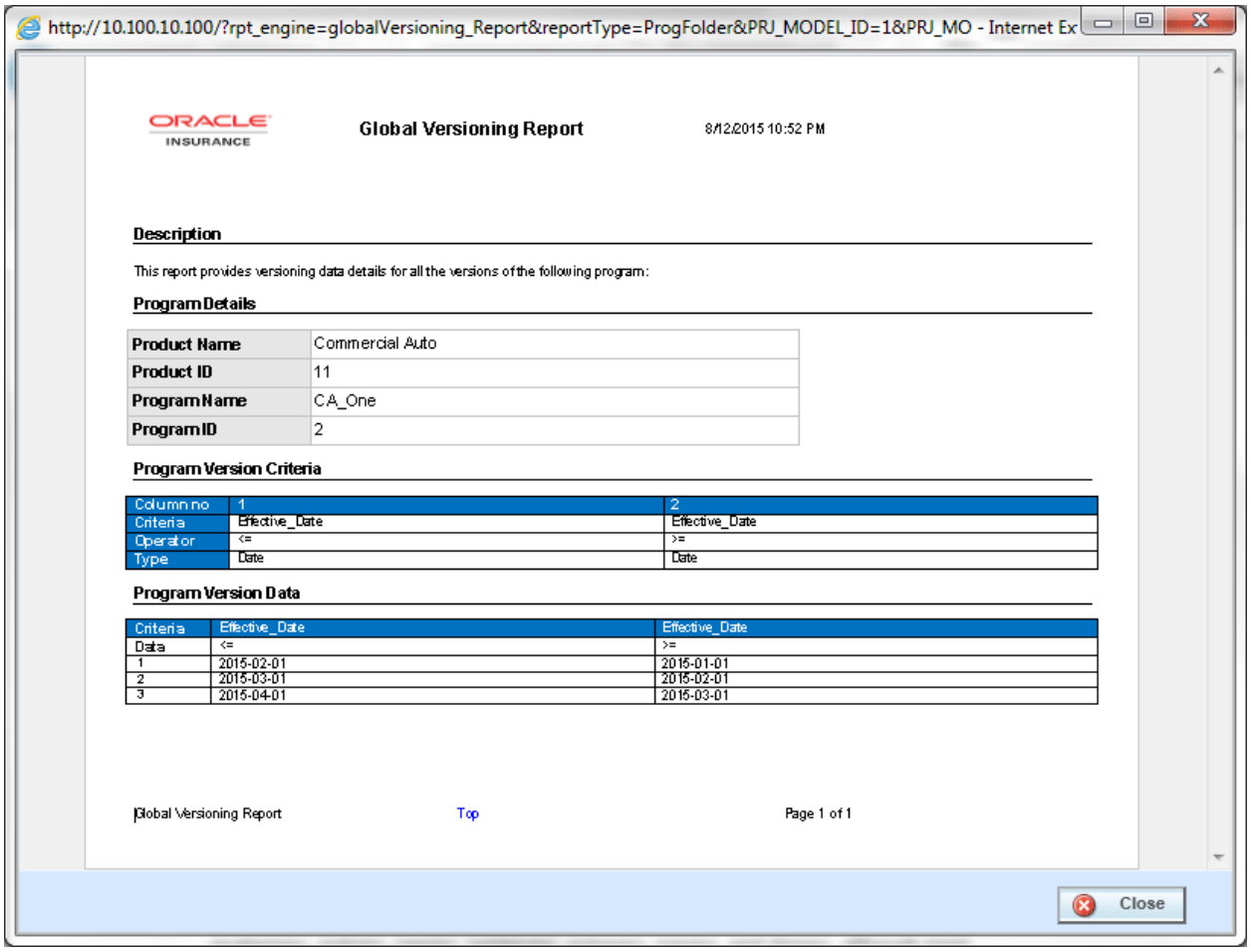


Figure 70 Global Versioning Report

Thread Settings

If you experience issues when generating a large number of reports, the system administrator can increase the performance by adjusting the reporting threads and global versioning report threads in the Preference settings. The use of threads is purely for performance. Reports should run faster as the optimal thread count is approached.

Administrator permissions are required for access the Preference settings.

Category Visibility

Category visibility allows you to select the categories that are available for use in the product. Categories are created and managed at the Universal level, which is found at the project level. This means that every category created will be available to every product in the project. This may result in categories being available to products that do not require that category.

For example, if the auto product and the home product are both in the project, you may not want to use a driver or vehicle category in any other product but auto. By using category visibility, you can remove the driver and vehicle categories from the home product.

Every category in the project is automatically available to every product. If you do not want a category to be available, you must remove the category visibility.

Using Category Visibility

Category visibility should be set prior to any programs being created. It is possible to remove a category or add a category back later on. The changes will only apply to any new or edited variables. Any variables using a removed category will continue to use that category.

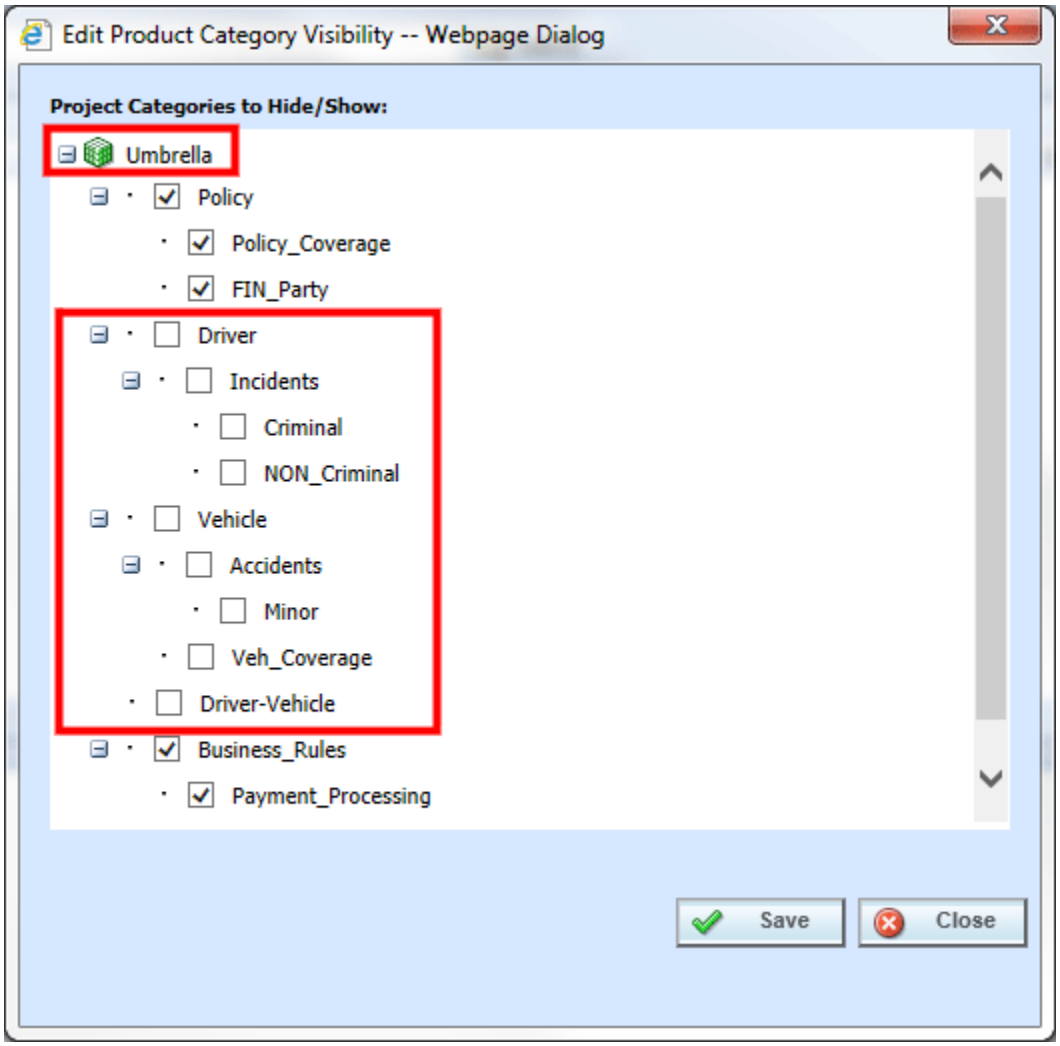


Figure 71 Selecting Category Visibility

1. Select the product where you want to set category visibility.
2. Right click and select Category Visibility.
3. On the category visibility screen, uncheck the categories that you do not want visible in the product. Any checked category will be available for use.
4. Click Save. A success message is displayed. Click OK.

5. Close the screen when you are finished. If you close the screen without saving, your changes will not be saved.

Product Right Click Menus

All products have the same right click menu options:

- **New Folder** – Opens the Manage Folders screen where you can create and manage folders for the product.
- **Create Versioning Package** – Create a versioning package for the product.
- **Global Versioning Report** – Creates a global versioning report.
- **Category Visibility** – Opens the Category Visibility screen where you can manage what categories are visible in the product.

If any of these options are grayed out or unresponsive, you may not have the proper permissions. Please contact your system administrator.

Chapter 7

FOLDERS

The RateManager file management system uses multiple folders and subfolders. This is a multilevel set up that allows you to place an unlimited number of folders and subfolders underneath a product. Using folders gives you a tree structure that can make it easier to locate a single program when there are a large number of programs to choose from. There are two types of folders:

- **Folders:** are the standard folders that reside under the product. These folders are visible at all times. Programs reside under folders. At least one folder must be created prior to any programs.

- **Program Version Folders:** are optional holding folders that reside under programs. These folders are visible only when a program is present. The program version folder structure is replicated under every program in the folder.

Folders

RateManager supports an unlimited number of folders and all folders are product specific. Folders can be edited and moved within the product where they were created. Each folder can contain multiple subfolders and/or multiple programs.

Folders allow you to classify and link programs together. This means you can group related programs underneath a major category and then continue to further define the structure until you reach the level where it no longer makes sense to subdivide.

For example, if you have programs that apply to a certain geographical area, you could create one top tier folder for all these programs. Within this folder you may want to create a subgroup that applies only to a specific region within this geographical area. Within the subgroup you may want to create a smaller subgroup that only applies to certain cities within this specific region. You can create as many folders and subfolders as you want and all these folders are linked together. There is no limit to the number of folders and subfolders you can create.

Program Version Folders

RateManager supports an unlimited number of program version folders and all program version folders are specific to the folder where they were created. Each program version folder can contain multiple subfolders and/or multiple program versions.

The program version folder structure is replicated in each program created in the folder. This means that if you create a program version folder called “archive” under the folder “North” then every program you create in the “North” folder will have a program version folder called “archive.”

Program version folders allow you to categorize program versions so you can quickly sort through multiple program versions. For example, if you have a program that had a new program version created each week you could create a program version folder to hold the program versions that were applicable for the first month, a program version for the second month and so on. This would allow you to quickly view the program versions that were for a specific month without having to scroll through a long list of program versions.

Manage Program Folders

Folders and program version folders are managed on the Manage Program Folders screen. On the Manage Program Folders screen, you can:

- Create a New Folder
- Rename a Folder
- Delete a Folder

Navigating to Manage Program Folders

Folders and program version folders are managed the Manage Folders option in the top bar menu on the Program Explorer tab. You must select the domain type, domain, project and product before the Manage Program Folders screen is shown.

1. In Program Explorer, select the product you need.
2. Click **Mange Folders** to open the Manage Program Folders screen.

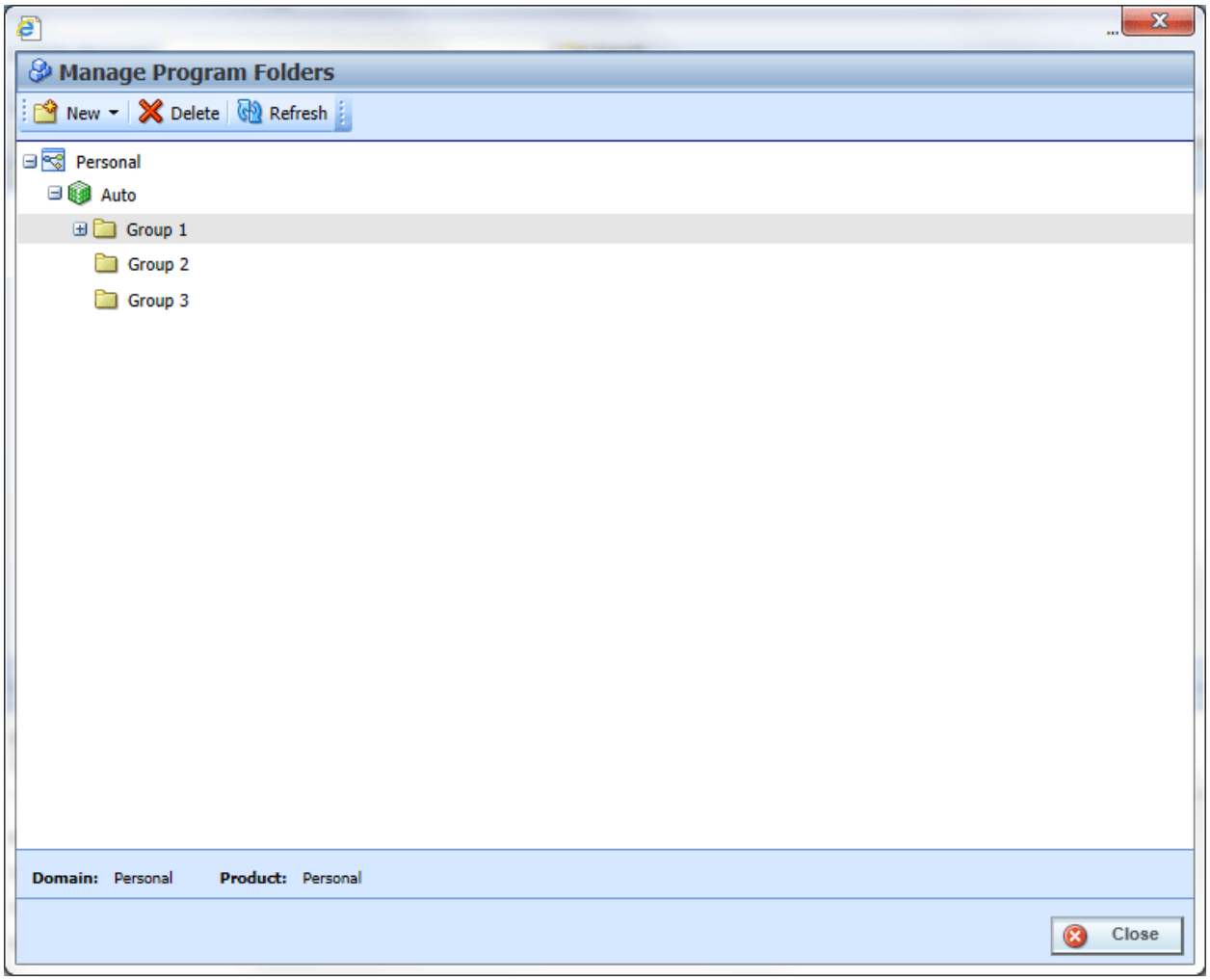


Figure 72 Manage Folders

MANAGE PROGRAM FOLDERS FUNCTIONALITY

The menu bar at the top of the screen contains the functions you can perform on this screen. If you do not have the rights to a certain function, it is not displayed. For example, if you do not have the rights to delete a folder, the delete option is not displayed.

New	**New:** Creates a new folder.
Delete	**Delete:** Deletes the currently selected folder.
Refresh	**Refresh:** Refreshes the screen with the current information.

MANAGE PROGRAM FOLDERS SCREEN DETAILS

The Manage Program Folders screen contains all current folders and program version folders for the selected product. Folders expand out into a tree view. Expand the product to see the folders underneath. At the bottom of the screen is an information bar that contains the domain and product you have selected.

Right Click Options

Every folder has a right click menu. To view the right click menu options, select the folder you want and right click. The menu options that you can perform depend upon the type folder selected. The options for a folder are different from the options available for a program version folder. Your ability to perform a function depends on your rights. For example, if you do not have the rights to delete a folder, you will receive a message.

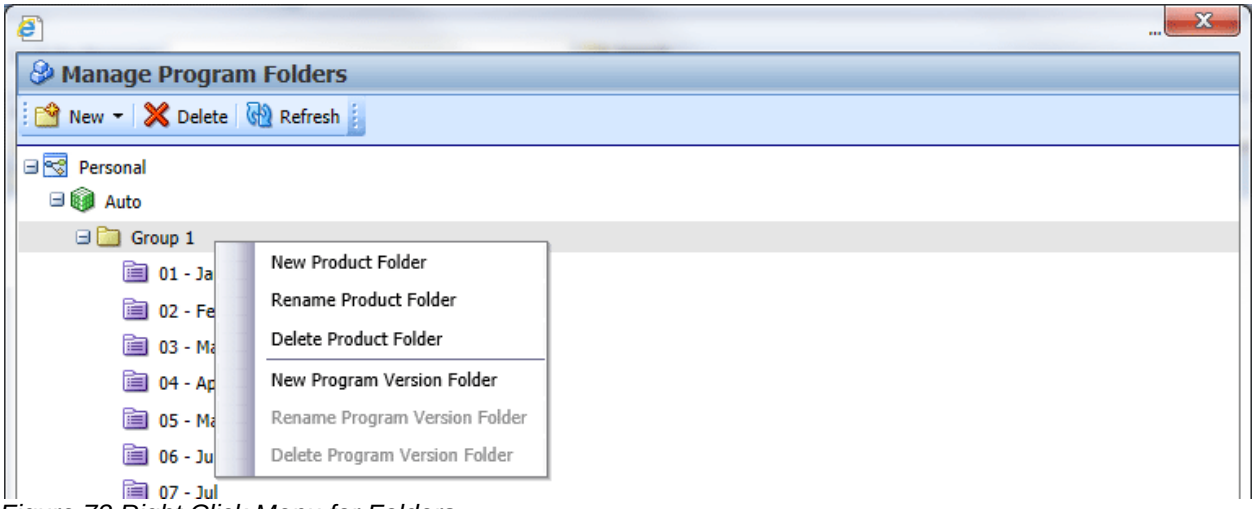


Figure 73 Right Click Menu for Folders

Manage Program Folders has six right click menu options available:

- **New Product Folder** – Creates a new folder under the folder you selected.
- **Rename Product Folder** – Renames the folder.

- **Delete Product Folder** – Deletes the selected folder. This option may not be active if you do not have rights.
- **New Program Version Folder** – Creates a new program version folder under the folder you have selected.
- **Rename Program Version Folder** – Rename the folder.
- **Delete Program Version Folder** – Deleted the selected folder, this option may not be active if you do not have the rights.

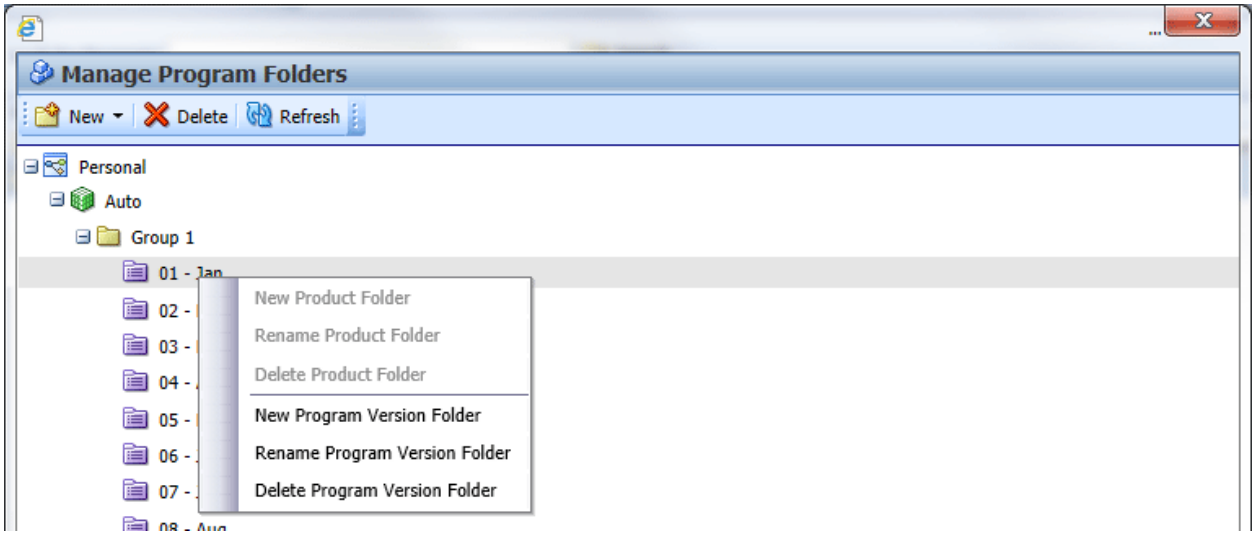


Figure 74 Right Click Menu for Program Version Folders

Folders have 4 options:

- New Product Folder
- Rename Product Folder
- Delete Product Folder
- New Program Version Folder

Program Version Folders have 3 options:

- New Program Version Folder
- Rename Program Version Folder
- Delete Program Version Folder

Program Version Folders cannot have product folders underneath.

CREATING A FOLDER

Folders have to be created before programs. If no folders have been created, the first folder created must fall under product. For any other folder, you can select the folder pathway.

1. If this is the first folder created, select the product. If another folder is available, you can create a new folder under it. If this is not the product you wanted, close the screen and return to the Program Explorer tab. Select the correct product and then open the Manage Folders screen.

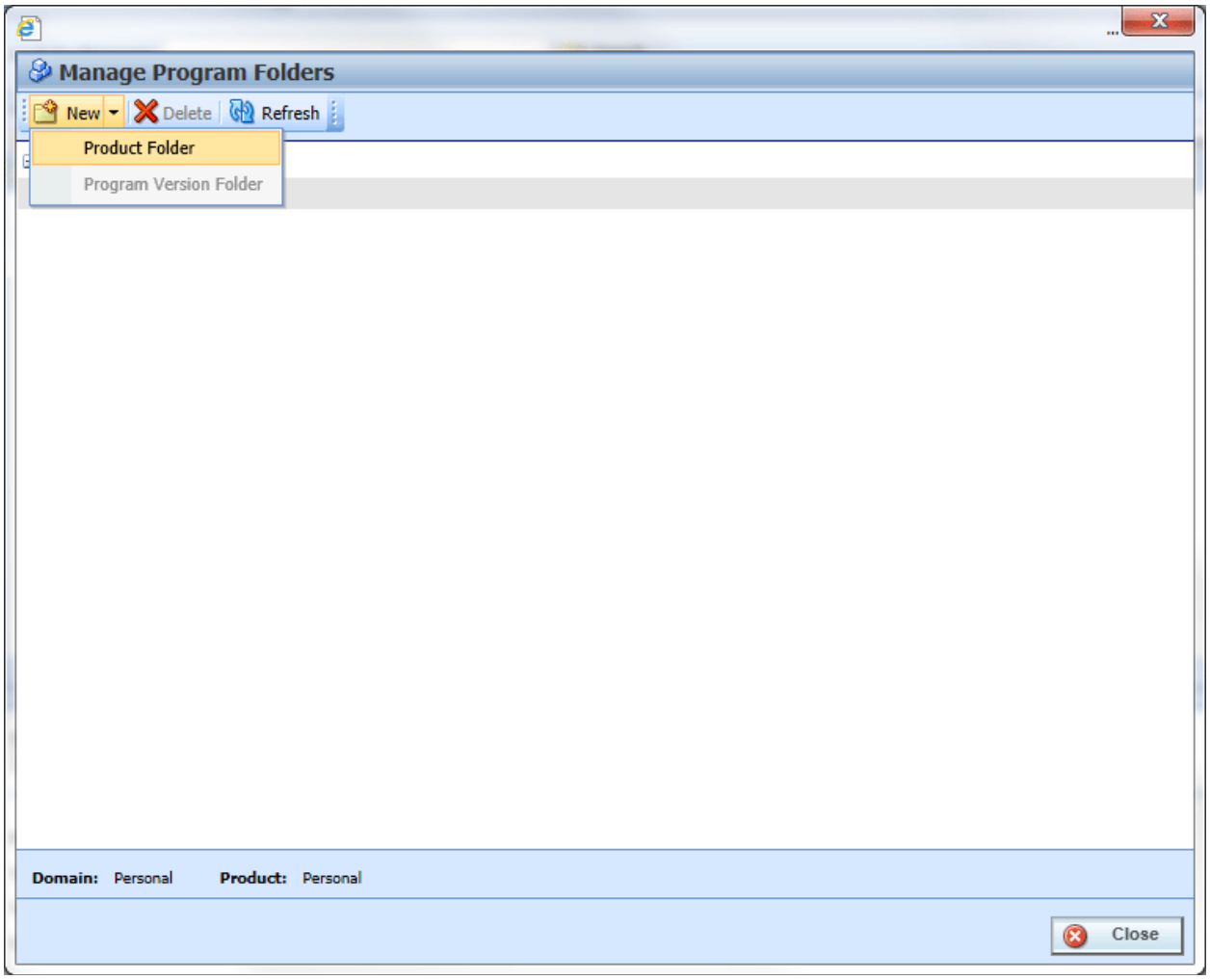


Figure 75 Creating New Folder

2. Select **New→Product Folder** from the top bar menu or right click and select New Product Folder from the menu.

3. A blank box is listed. Enter in a **Name**.

4. Click out of the box to exit the field and save your entry. The screen refreshes and the new folder is displayed in the Manage Program Folders screen.

5. Refresh the Programs area of Program Explorer to view the new folder.

RENAMING A FOLDER

Folders can be renamed at any time in the Manage Program Folder screen or on the Program Explorer by selecting a folder, right clicking and selecting the Edit Folder feature.

1. Select the folder you want to rename and right click.

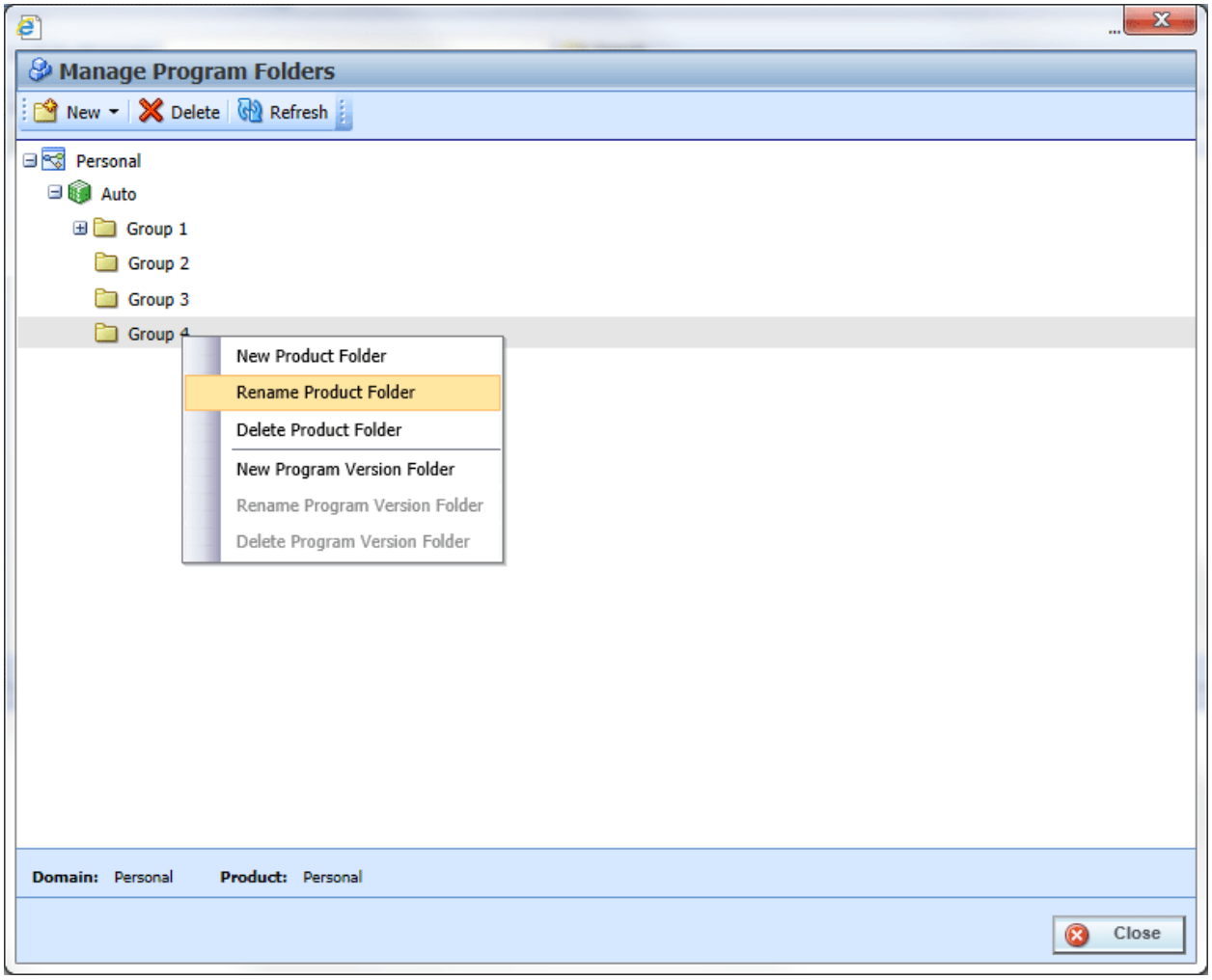


Figure 76 Renaming a Folder

2. Select **Rename Product Folder** from the menu.

3. The folder name is boxed, indicating the field is open for editing. Enter in the new name.

4. Click outside the box to exit the field and save your entry. The screen refreshes and the folder is displayed in the Manage Program Folders screen.

5. Refresh the Programs area of Program Explorer to view the updated folder.

NOTE: *Oracle Insurance recommends that the Folder Name be unique. For example, in Figure 76 Renaming a Folder, no other folder under the Auto product, should be called Group 4. Names cannot be duplicated at the same folder level.*

DELETING A FOLDER

If a folder is no longer needed, it can be deleted. Make absolutely sure that the programs for the folder are no longer needed. Deleting a folder deletes all programs and program version folders under that folder only. Subfolders and any programs in the subfolders are not deleted but are now unlinked. Folders that are unlinked are no longer accessible or visible in Programs on the left side of Program Explorer. Unlinked folders can be viewed in the Manage Program Folders screen.

To access a subfolder whose parent folder has been deleted, you have to move the subfolder to a visible folder. All visible parent folders are located on the left side of Program Explorer. Any subfolders underneath are moved as well. It is recommended that any subfolders either be moved or deleted prior to deleting the parent folder. You also can delete a top tier folder to unlink all subfolders rather than deleting folders.

1. In the Manage Program Folders screen, select the folder you want to delete and right click. Select **Delete** or right click and select **Delete Product Folder** from the menu.

2. You are asked to confirm deletion of the folder.

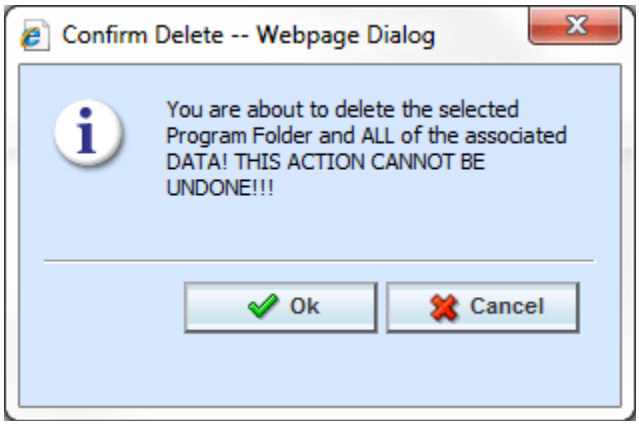


Figure 77 Deleting a Folder

3. Select **OK** to confirm deletion or **Cancel** to return to the Manage Program Folders screen without deleting the folder.

4. You are asked again to confirm deletion of the folder.

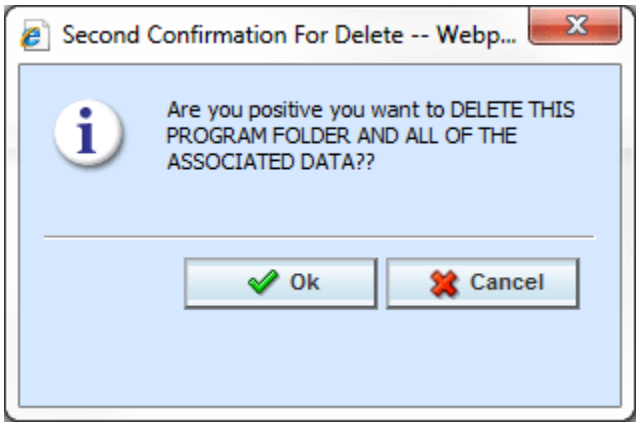


Figure 78 Confirming a Folder Delete

5. Again, select **OK** to confirm deletion. The folder and all associated programs will be deleted. Click **Cancel** to return to the Manage Program Folders screen without deleting the folder.

Right Click Folder Options

On the Program Explorer tab, the Programs area and the Program Detail area both have right click menus on folders. Folders located in the Programs area of Program Explorer have 4 right click menu options. Folders in the Program Detail area of Program Explorer have 2 right click menu options.

The Edit Folder option is only found in the right click menus on the Program Explorer tab.

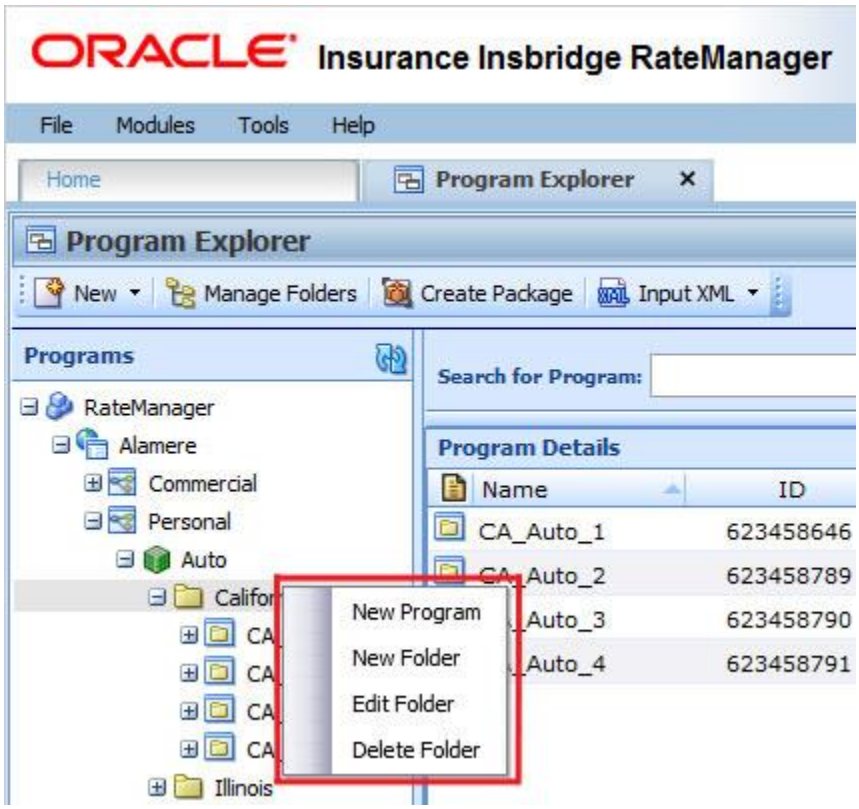


Figure 79 Folder Right Click Menu

- **New Program** – Creates a new program under the selected folder.
- **New Folder** – Opens a New Folder screen where you can create a new sub folder under a selected folder. Or create a new folder under the product. A new folder can also be created by right clicking a Product and selecting New Folder.
- **Edit Folder** – Opens the Edit Folder screen where you can edit the selected folder name and move the folder. To move a selected folder, you must use the Edit Folder option on the right click menu of the folder.
- **Delete Folder** – Deletes the selected folder and all contents.

EDITING A FOLDER

Edits to a folder can change the folder name or move the folder to a different location within the same product. You cannot move a folder outside of the product where it was created. When you change the pathway of any folder, the folders underneath are changed as well.

If you have multiple folder name edits, use the Rename Product Folder option on the Manage Folder screen. That screen allows for any folder name to be changed. If you want to edit a specific folder name, use the right click option, Edit Folder.

Program Version Folders names can be edited on the Manage Folder screen only.

1. In the Program Explorer tab, select the folder you want to edit and right click. Select Edit Folder.

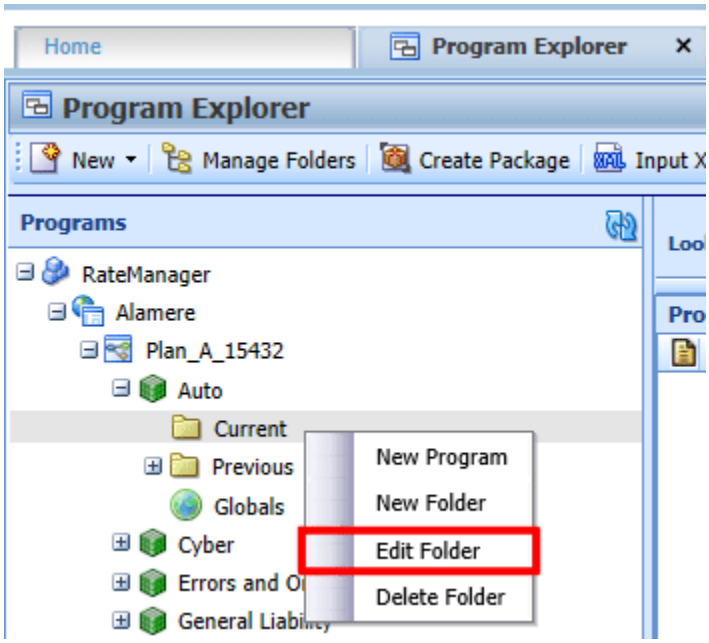


Figure 80 Folder Options Found on the Right Click Menu of Programs

2. The Edit Folder screen is shown.

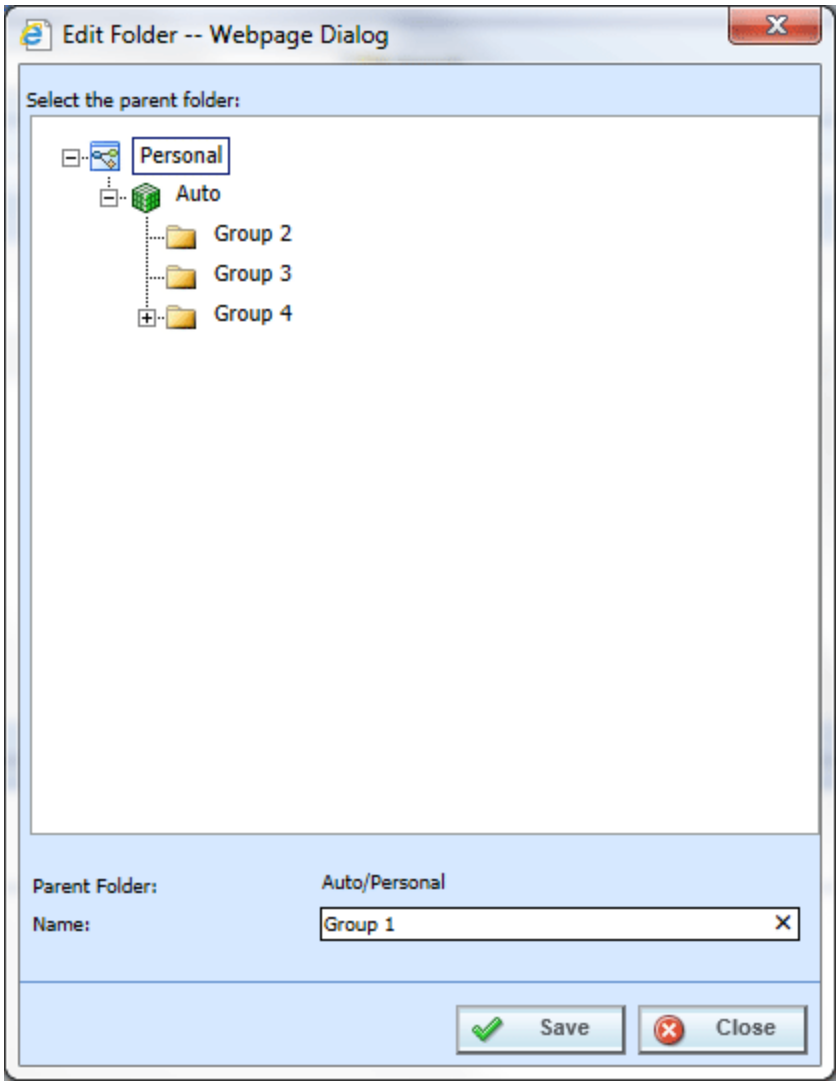


Figure 81 Edit Folder

3. Make your changes and click Save. The Edit Folder screen closes returning you to the Program Explorer tab. Click Close to go back to the Program Explorer tab without saving changes.

4. If your changes are not presenting, click File→Reload to refresh the screen.

MOVING A FOLDER

Moving a folder changes the folder location within the same product. You can move any folder either up or down the folder structure. When you move a folder, all associated programs and subfolders are moved as well. There is no limit to the number of times a folder can be moved. Moving is performed on the Edit Folder screen.

Program Version Folders cannot be moved.

1. In the Program Explorer tab, select the folder you want to move and right click. Select **Edit Folder**.

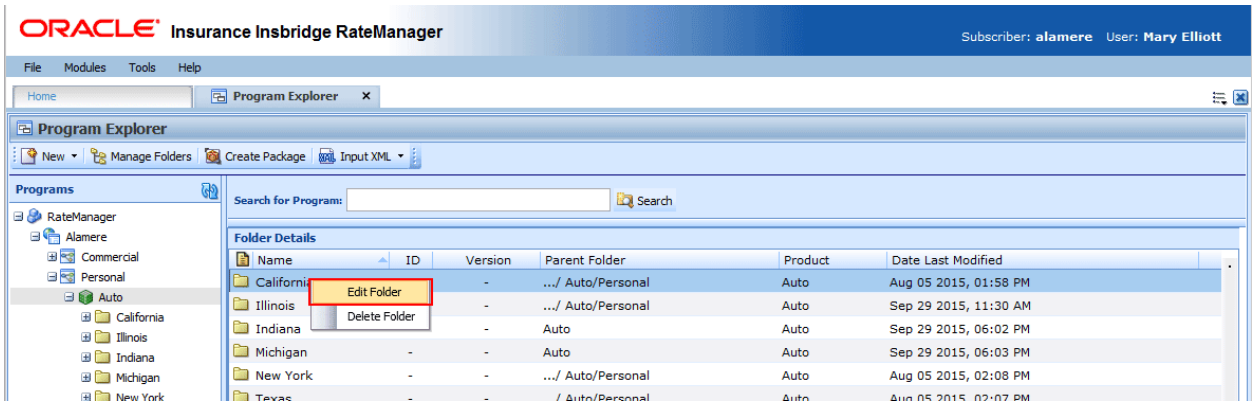


Figure 82 Folder Options Found on the Right Click Menu of Program Details

2. The **Edit Folder** screen is shown. The folder you have selected to move is displayed in the Name field but not in the tree structure. Double click the destination folder where you want to move to. The Parent Folder field reflects the new path for the folder.

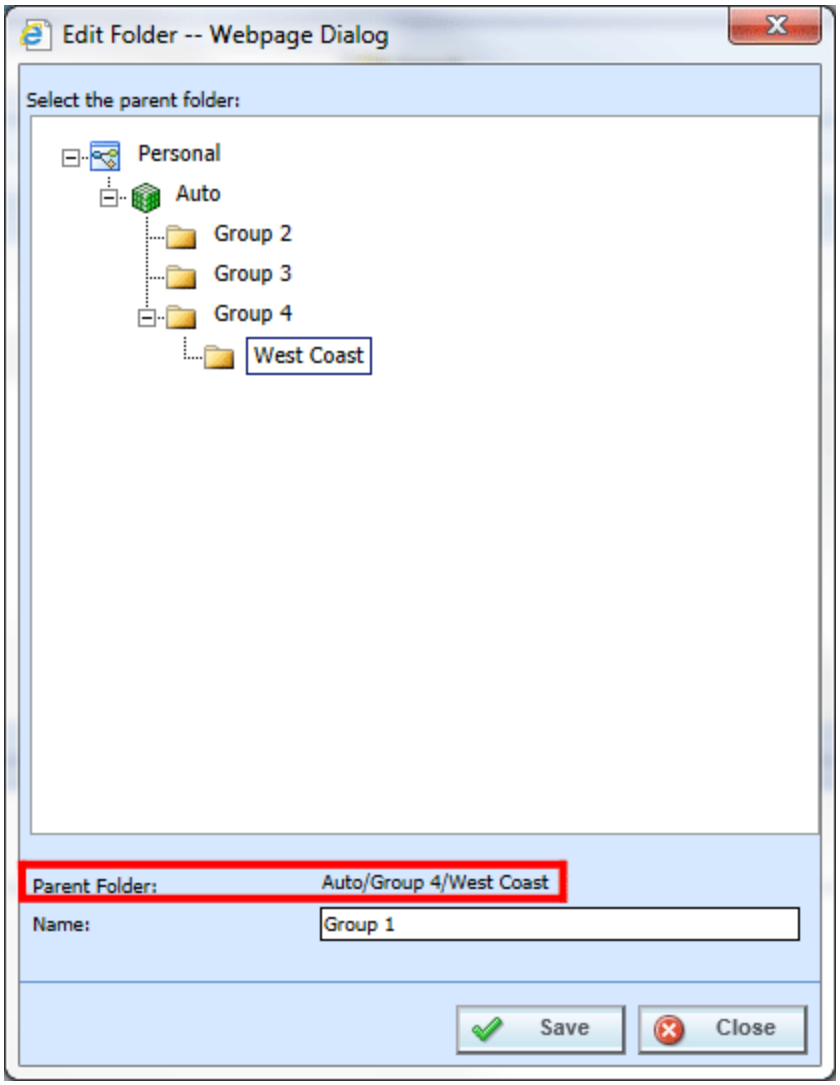


Figure 83 Moving a Folder

3. If this is correct, click **Save**. The Edit Folder screen closes, returning you to the Program Explorer tab. Any subfolders are moved as well. Click **Close** to go back to the Program Explorer tab without saving changes.

4. If your changes are not presenting, click File→Reload to refresh the screen.

CREATING A PROGRAM VERSION FOLDER

Program Version Folders can be created only on the Manage Program Folders screen. At least one folder has to be created before a Program Version Folder can be created. Program Version Folders are listed in alphabetical order.

1. Select the product where you want to create a program version folder and click Manage Folders.

2. Select the folder where you want to create a new program version folder. Make sure this is where you want the folder. Program Version folders cannot be moved.

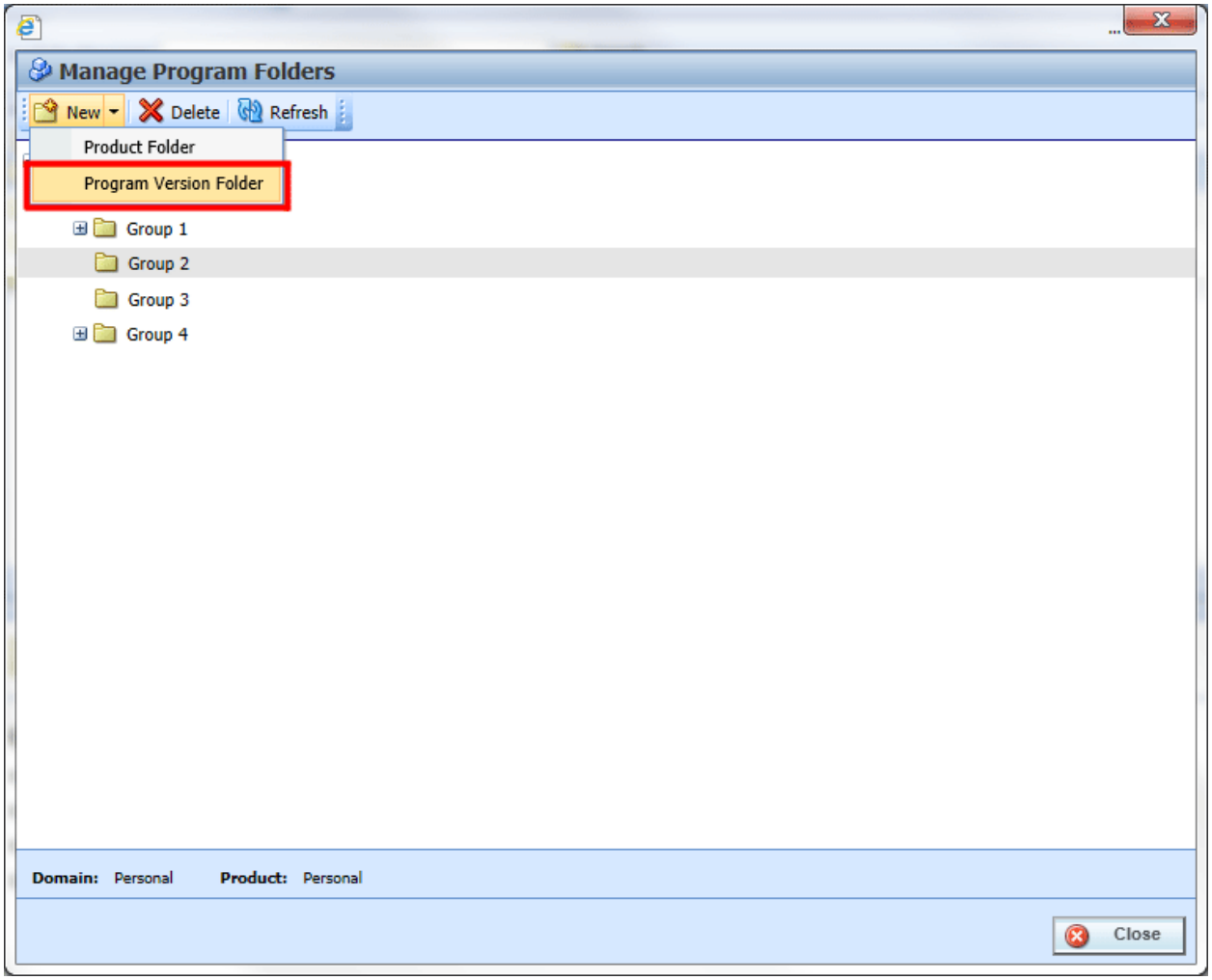


Figure 84 Creating New Program Version Folder

3. Select New→Program Version Folder from the top bar menu or right click and select New Program Version Folder from the menu.

4. A blank box is listed under the folder you selected. Enter in a Name.

5. Click out of the box to exit the field and save your entry. The screen refreshes and the new folder is displayed in the Manage Program Folders screen.

Program Version Folders are displayed under programs. New Program Version folders may not be

displayed on the Program Explorer tab until a program has been created at the same level.

RENAMING A PROGRAM VERSION FOLDER

Program Version Folder can be renamed at any time in the Manage Program Folder screen.

1. Select the program version folder you want to rename and right click.

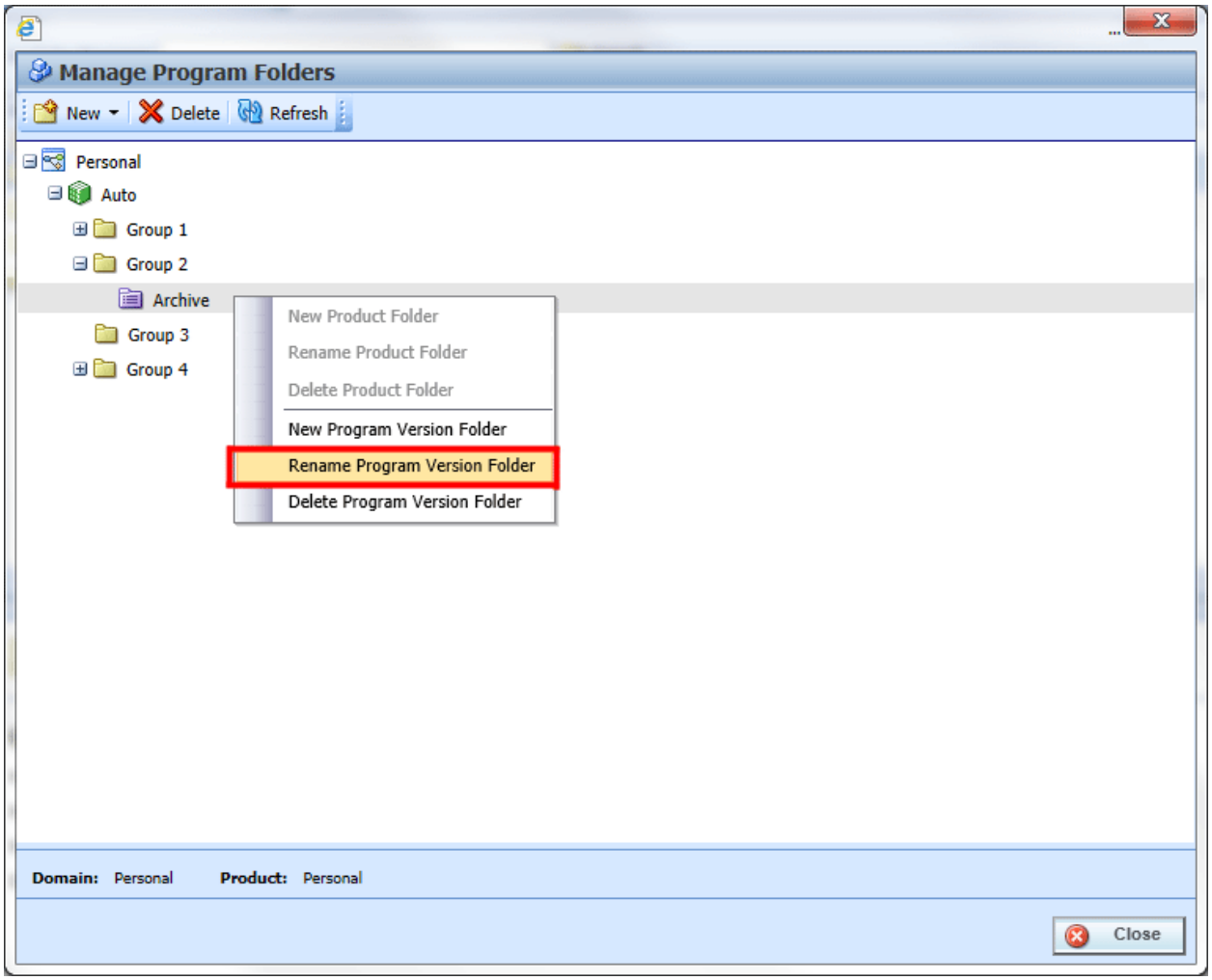


Figure 85 Renaming a Folder

2. Select Rename Program Version Folder from the menu.

3. The program version folder name is boxed, indicating the field is open for editing. Enter in the new name.

4. Click out of the box to exit the field and save your entry. The screen refreshes and the folder is displayed in the Manage Program Folders screen.

NOTE: *Oracle Insurance recommends that the Folder Name be unique, regardless of the pathway. Names cannot be duplicated within the same folder level.*

DELETING A PROGRAM VERSION FOLDER

If a program version folder is no longer needed, it can be deleted. Be aware that when you delete a folder, programs do not get deleted. Programs are moved to the parent folder. Any program version subfolders are removed.

1. In the Manage Program Folders screen, select the program version folder you want to delete and right click. Select Delete Program Version Folder.

2. You are asked to confirm deletion of the folder.

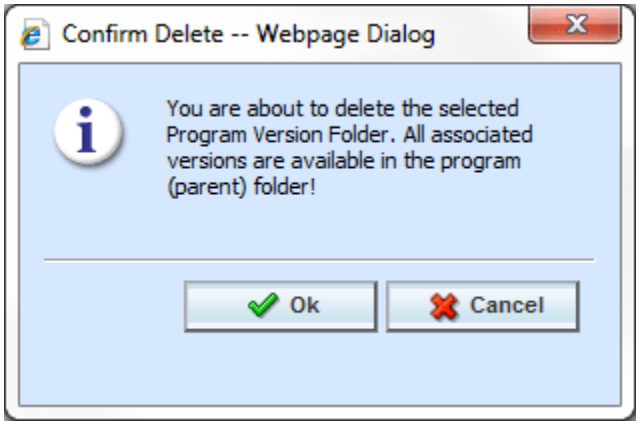


Figure 86 Deleting a Program Version Folder

3. Select OK to confirm deletion or Cancel to return to the Manage Program Folders screen without deleting the program version folder.

4. The program version subfolder is deleted.

PROGRAMS

Programs in RateManager typically correspond to rate manuals. For example, if you had rate manuals for California and New York auto, then you could have corresponding RateManager programs for California and New York under the auto product. When a rate change occurred, you would create a new version of the appropriate program.

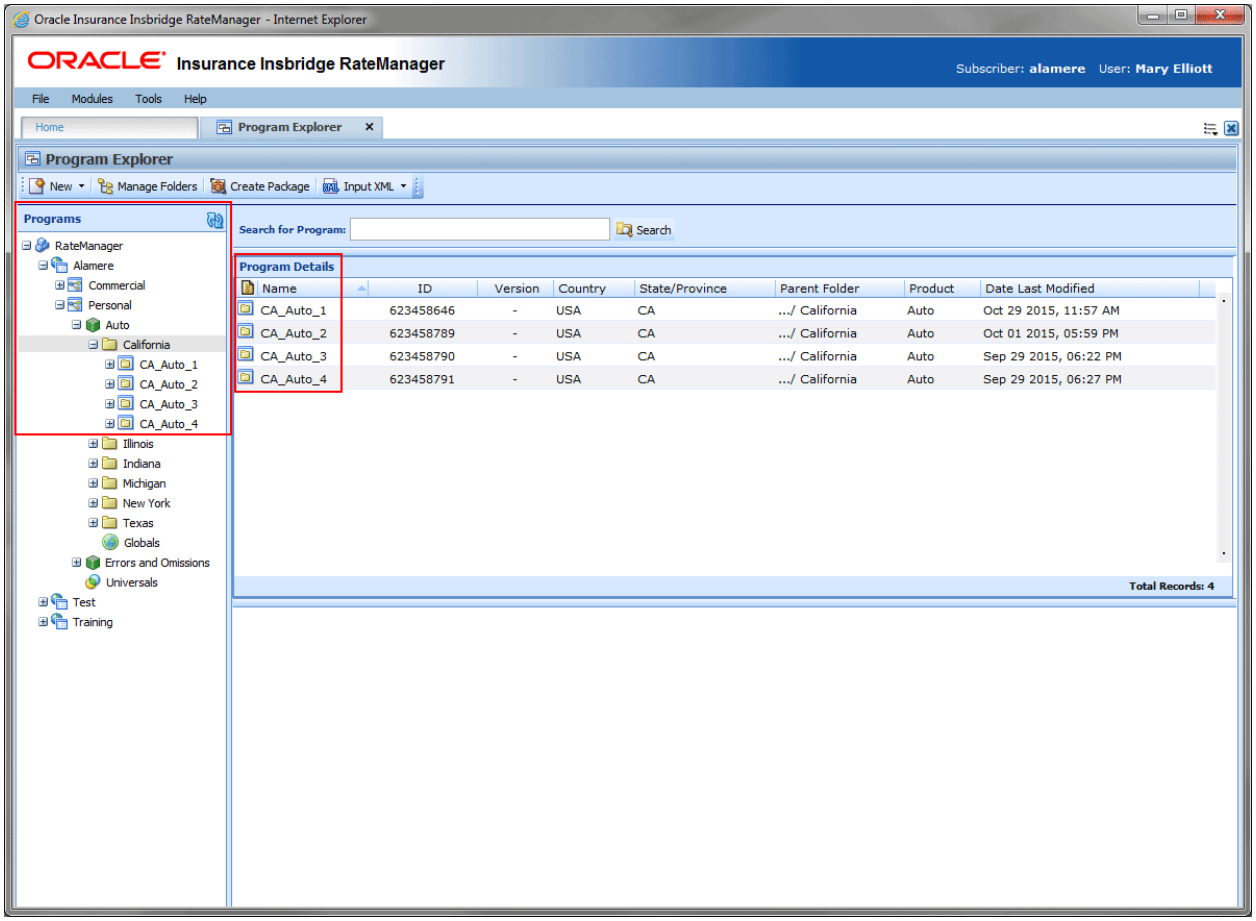


Figure 87 Programs on Program Explorer

Programs are created and managed in Program Explorer. Programs are displayed in the Programs area when you expand a folder. Programs are listed in the Program Detail area when you select a folder. From either area you can:

- Create a New Program
- Move a Program
- Delete a Program
- Edit a Program
- Manage Version Criteria
- Run a Program Version Report, and Program Difference Reports.
- Map Callouts

There are two types of programs:

Standard Programs These are programs that you created. Unless locked, you can add, edit or delete any item from this program. Standard programs can be created under any domain type.

Template Programs These are programs that were created from a template. Template programs automatically have variables, algorithms and driver assignments locked. These items cannot be edited or deleted. You can add items and create revisions. Any globals or universals associated with a template are locked as well.

The RateManager domain type only contains standard programs. You will not be allowed to apply any template programs into the RateManager domain type.

Template programs can only be under an Other domain type only. Other domain types also can contain standard programs.

RateManager supports an unlimited number of programs and all programs are product specific. For example, creating **Program 1** for **Carrier A** under the Auto product does not create **Program 1** for **Carrier A** under another product.

Versioning Criteria

Versioning Criteria is used when you want to define which version of a program you want to use based on a policy level field.

FOR MORE ON VERSIONING CRITERIA, PLEASE SEE VERSIONING CRITERIA

Versioning Criteria is used when you want to define which version of a program you want to use based on a policy level field. For example, if you have a program with three versions, you can select which version to use by the effective date (a policy level field). You can use version one if the effective date is this month. Version two can be used if the effective date is next month and version three can be used for any other date.

Setting up versioning criteria is a two-step process:

Step 1 – At the program level, select the criteria you want to use. This is the field that determines which version of the program should be used.

Step 2 – At the program version level, you enter the defining parameters of the field.

If you want to use this feature, you must create a global versioning package. A global versioning package allows for version criteria to be viewed at the program level. Please see Creating a Global Versioning Package for more information.

NOTE: *Only Policy level fields can be used as Version Selection Criteria.* The *maximum number of criteria that can be used is 10.*

Step 1 Select the Versioning Criteria

1. Select the program where you want to add versioning criteria and right click. Select Program Properties.
2. This opens the **Edit Program** screen where manage version selection criteria.
3. Select the criteria in the order you want them run. Or make any changes to your selections.
4. When you have finished click **Save** to save your selections or edits. Click **Close** to close the Edit Program screen and return to Program Explorer.

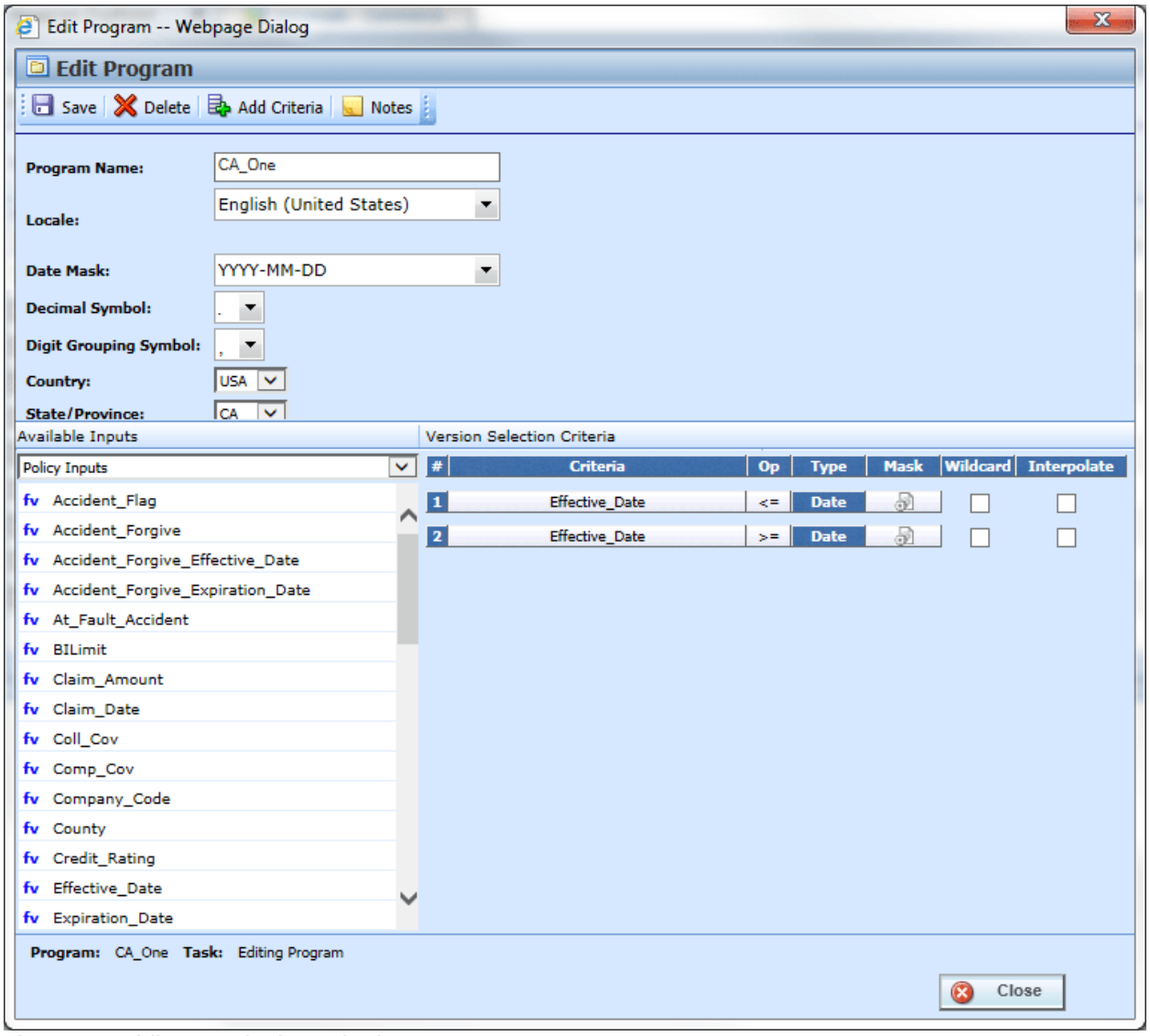
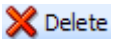
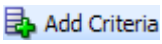
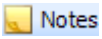


Figure 99 Adding Versioning Criteria to a Program

Versioning Criteria Functionality

The **Menu Bar** at the top of the screen contains the functions you can perform on this screen.

	Save: Saves your entries.
	**Delete:** Deletes the currently selected field criteria.
	**Add Criteria:** Adds the selected field as criteria.
	**Notes:** Enter a note. Please see Notes for more information.

Versioning Criteria

The Edit Program screen contains the program information in the top portion of the screen and the versioning criteria in the lower portion of the screen. There is also an information bar at the bottom of the screen.

Program Information:

Program Name – Name of the program. This field can be edited at any time.

Locale – The locale used in the program.

Date Mask – The date mask used by the program. For more information on date masks, please see
Date Masks.

Decimal Symbol – The decimal symbol to be used in the program.

Digit Grouping Symbol – The digit grouping symbol to be used in the program.

Country – The country where this program applies.

State/Province – The state or province where this program applies.

Versioning Criteria: Available Fields

The fields available for selection are listed. You also can create a new policy level field by scrolling to the bottom of the Available Fields list and clicking <New Field Variable>. Please see Adding a New Field from Another Area for more information.

Versioning Criteria: Version Selection Criteria

 # – Sequential order of criteria selected from **Available Fields**.

Criteria – Displays the name of each field selected as criteria for this program.

Op (operator) – Determines how the data in the table is used. The default operator is an equals sign (=). To change the operator, select the current operator, right click and choose a new operator. See Comparison Operators for more details.

NOTE: *Arithmetic comparison functions such as < and > cannot be used on string values. Only the operators of equal and not equal are available for string variables.*

Type – Shows the data type of the criteria. The type of data in the field must match that of the criteria or an error will occur.

Mask – Determines how the data being passed into the program should be interpreted by allowing you to hold, remove or replace characters being passed. (For example: If you only wanted to accept the first, second and third characters of a VIN number being passed, this feature adds the ability to hold these characters and disregard the rest.) Click the icon to set the mask. See Masking for more information.

Wildcard – A check in this field tells the system that one or more rows will accept any value passed for these criteria. Exact matches are not required. See Wildcard for more information.

Interpolate – A check in this field instructs RateManager to use interpolation to determine the correct value for the table variable. See Interpolation for more information.

Information Bar

At the bottom of the screen is an information bar that contains the program name and the task you are performing.

Right Click Options

The Criteria and Op fields have a right click menu. To view the right click menu options, select the field you want and right click. All menu options are displayed.

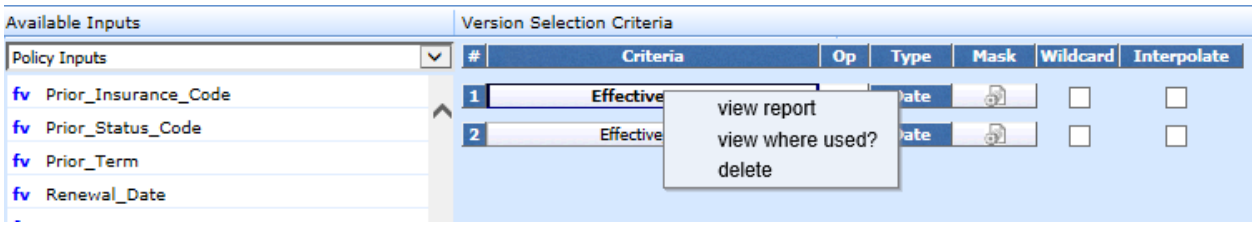


Figure 100 Right Click Menu for the Setting Versioning Criteria

The **Criteria** column has three right click menu choices:

- **View Report –** Displays field information; XML ID, Type, and Category.
- **View Where Used –** Displays the View Where Used report. Please see View Where Used for more information.
- **Delete –** Deletes the selected criteria.

The **Op** column has six right click choices:

- **Equals**
- **Not equals**

- **Less than**
- **Less than or equal to**
- **Greater than**
- **Greater than or equal to**

Setting Versioning Criteria

1. Select the field you want to use from the listing of **Available Fields**. You can either select the field and click **Add Criteria** or double-click the field. The field is displayed in the Version Selection Criteria area. A total of 10 criteria can be selected.

2. Set the comparison operator to be used when interpreting data by right clicking the operator box in the Op column (the default is an equals sign =). See Comparison Operators for more information.

3. Make your entries if you want to enter a mask, wildcard or interpolate.

4. Click **Save** to save your work. Click **Close** if you are finished entering criteria.

Deleting Versioning Criteria:

1. To remove a field as criteria, click the name in the Version Selection Criteria listing and select **Delete** from the menu bar or right click the criteria and select Delete from the menu.

2. Click **Save** to save your work. Click **Close** if you are finished editing criteria.

NOTE: *If you do not click Save your work will not be saved automatically when you exit the screen.*

Step 2 Entering Versioning Criteria

Versioning criteria are per program version.

1. Select the program version where you want to add versioning criteria and right click. Select **Edit Versioning Data**.

2. This opens the **Edit Program Version** screen where you can update the versioning data information used by the selected criteria fields. Fields cannot be added on this screen. If a field needs to be added or removed, you must return to Edit Program screen.

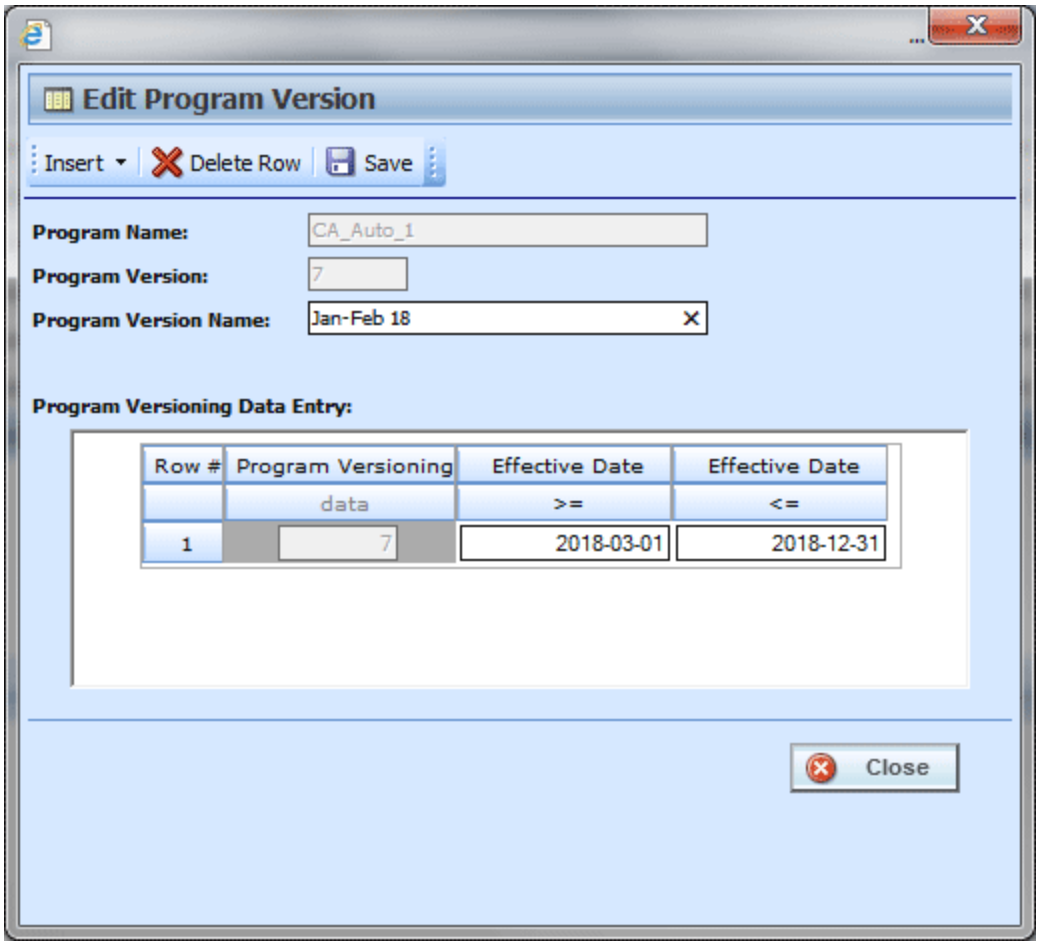


Figure 101 Editing a Program Version

3. When you have finished click **Save** to save your selections or edits. Click **Close** to close the Edit Program Version screen and return to Program Explorer. If you do not save prior to closing, your updates will not be saved.

Program Versioning Data.

Creating a Program

New programs can be added at any time to any folder in any domain type. There is no limit to the number of programs you can enter.

1. In Programs, select the domain type, domain, project, product and folder where you want to create a new program.

2. Click New Program from the top bar menu or right click the folder and select New Program. The New Program screen opens, allowing you to enter the program name, select a locale, select a date mask, decimal symbol, digit grouping symbol and select a country and state/province.

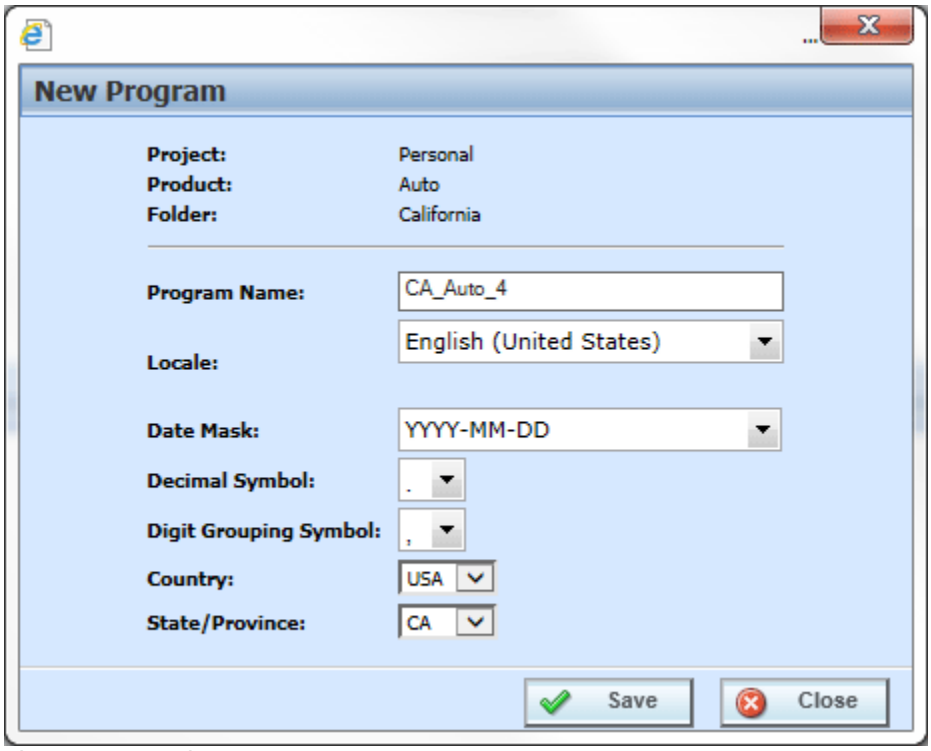


Figure 88 Creating a New Program

3. Click Save to save your entry and close the window. The screen refreshes with the program information you entered. If you double-click the new program name, you will see that Version 1 has been created for you.

The Locale, Date Mask, the Decimal Symbol and Digit Grouping Symbol are automatically filled with the defaults from the product. The values set in the product apply as the defaults to every program created in the product. The program can select a different locale if necessary. For more on local, decimal and digit grouping symbols please see RateManager Constructs.

WARNING: *Changing date and number formats may result in programs not running properly. Global variables that have been created as one type of date and number format may not function as expected with another date and number format. Date and number formats should be consistent through the program.*

Moving a Program

You can move a program to any folder under the same product where the program is located. Moving a program moves the program and all versions. Program moves can be done from the Programs area or the Program Detail area.

1. Select the program you want to move.

2. Right click and select **Move Program**. A **Move Program** screen is shown.

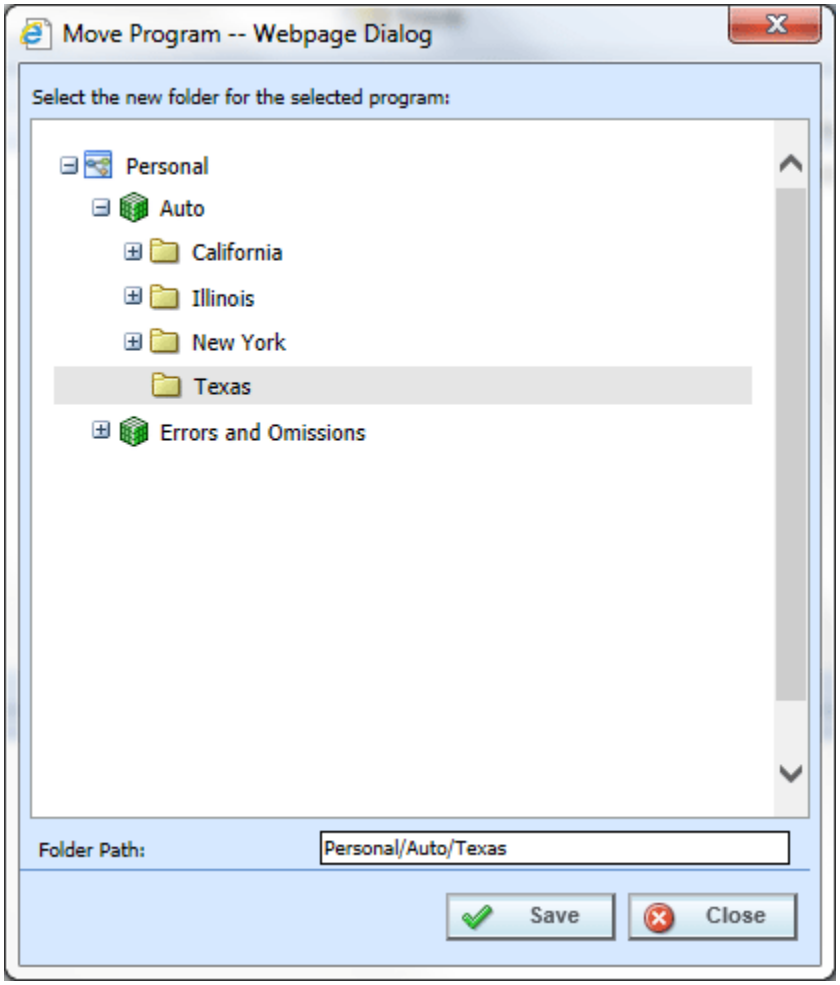


Figure 89 Selecting Destination for Moving a Program

3. The Folder Path at the bottom of the window displays the current location of the program. Click the folder where you want to move the program to. The Folder Path reflects the final destination for the folder.

4. When you have the program in the correct folder, click **Save**. The screen refreshes and the program is moved to the selected folder.

5. If your changes are not presenting, click File→Reload to refresh the screen.

Editing a Program

Program information can be edited at any time. Use the Program Properties option when you want to change the name, date mask, country or state/province of a program or select which inputs are to be used for program version selection.

1. Select the program you want to edit and right click and select **Program Properties**.

2. This opens the **Edit Program** screen where you can change the program name, locale, date mask, decimal, digit, country, or state/province. You also can add version selection criteria.

3. When you have finished click **Save** to save your changes. Click **Close** to close the Edit Program screen and return to Program Explorer.

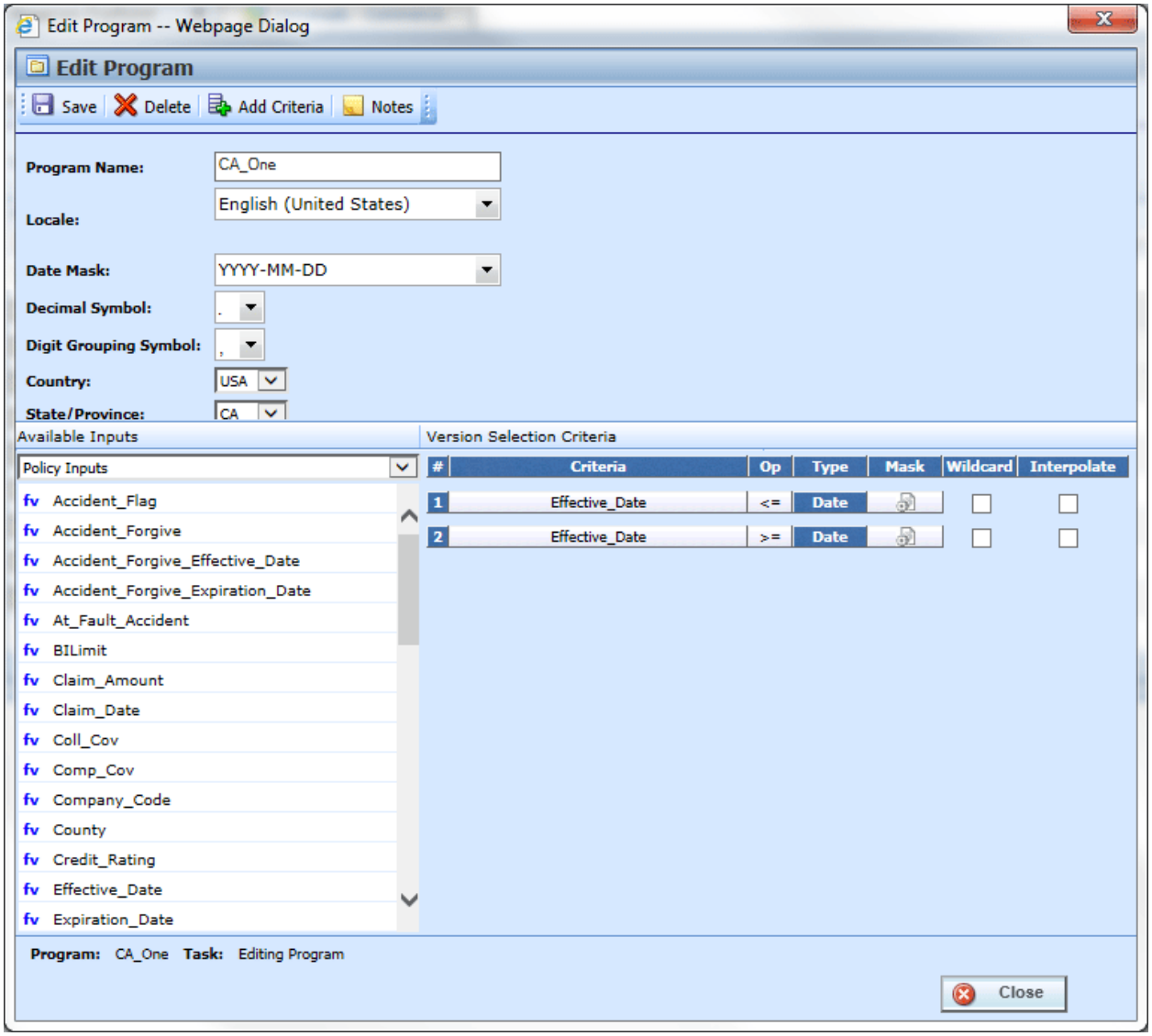


Figure 90 Editing a Program

Deleting a Program

If a program is no longer needed it can be deleted. Make absolutely sure a program is no longer needed before deleting it. When a program is deleted, all data (variables, algorithms, sequencing, result groups and, if an auto program, driver assignment scenarios) and versions are deleted.

NOTE: *Program deletion cannot be undone.*

1. Select the program you want to delete.
2. Right click and select **Delete**. You must confirm the deletion of the program.

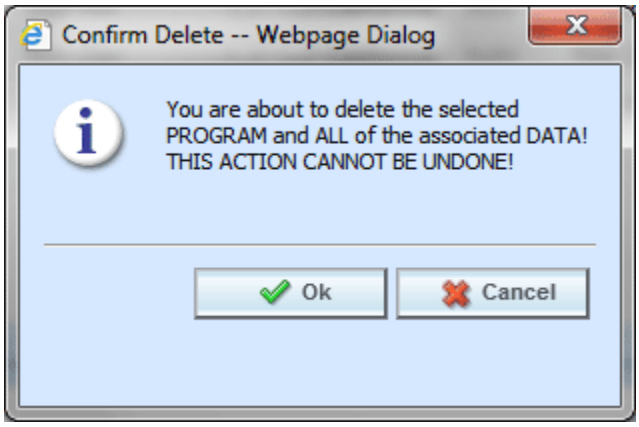


Figure 91 Delete Warning Message

3. Select **Ok** to confirm deletion or Cancel to return to Program Explorer without deleting the program.
4. You must confirm again that you want to delete the program.

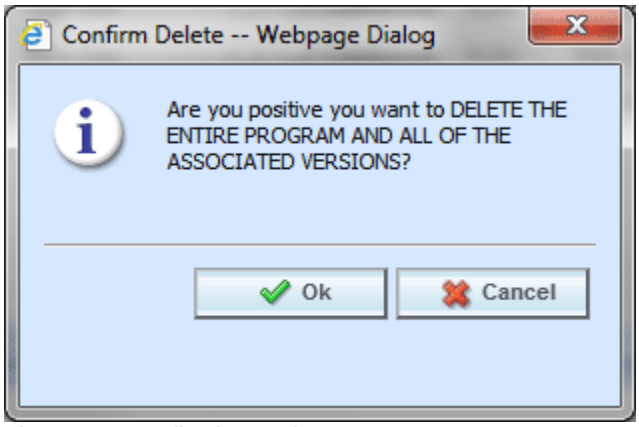


Figure 92 Confirming Delete

5. Select **OK** to confirm deletion. The program and all associated versions are deleted. Click Cancel to return to Program Explorer without deleting the program.

PROGRAM VERSIONS

Versioning is handled at the program level. When a new version of a program is created, it is initially a pointer to all elements in the last version of the program. When a change is made to an element in that version, it can be made version specific or applied to multiple versions.

Each program created in RateManager automatically starts at Version 1. When you create a program, version 1 is created at the same time. New versions of a specific program can be created at any time. Usually new program versions are created to correspond to a rate change.

Under a program, all program versions contain the same variables, algorithms, sequencing, mappings, and globals. Program versions differ when you select different revisions of these elements. For example, under Program A, you have two program versions. In version 2, you have changed a table variable to use revision 2 of that table variable and you changed an algorithm to use revision 4 of that algorithm. Only program version 2 will change. Version 1 continues to use the same revisions. The same variables and algorithms are in both program versions, but by selecting different revisions of these elements, you have changed program version 2.

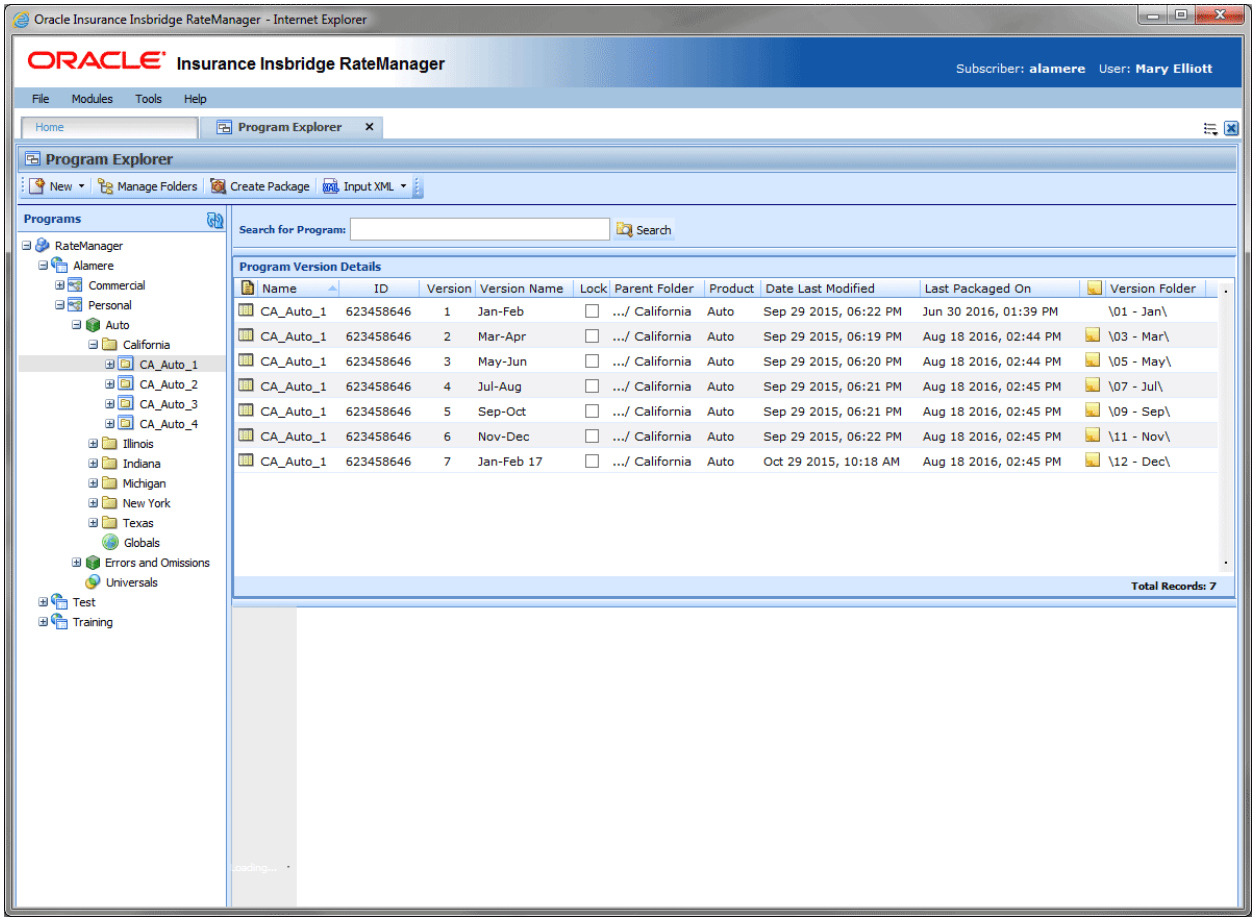


Figure 93 Program Versions in Program Explorer

Program versions are created and managed in Program Explorer. Program versions are listed in the Program Details area when you select a program. From here you can:

- Create a New Program Version
- Copy a Program Version
- Delete a Program Version
- Edit a Program Version
- Manage Versioning Data
- Assign program versions to Program Version Folders
- Unassign program versions to Program Version Folders
- Test Program, please see Test Program
- Map Callouts, please see Callouts
- Map Custom XML, please see Input and Output Mapping
- Notes, please see Notes
- Run a Program Version Report, please see Program Version Report
- For Template programs you will be able to run a Reconciliation Report, please see Reconcile

PROGRAM VERSION TYPES

Program versions can be created by you under any program in any domain type or under a template generated program in an Other domain type. Program versions can also be brought in under a template generated program in an Other domain type. Different icons can help you to visually identify the program versions that were applied from a template from the ones that you created.

User created program versions can be created or imported or versioned from a template program. Most often, user created program versions are displayed in the RateManager domain type but may be present if you create a new program version in a template generate program in an Other domain type.

When you copy a template generated program, version 1 is displayed with a template created icon. This is to designate that the program and version 1 were created from a template. A copied template generated program can accept additional versions of the program that may be entered in the library. You can create a new program version underneath the template generated program. Those program versions are displayed with a user created icon. You also can apply another template version from the library, which will be displayed with the template created icon.

Copied template program versions are displayed with a user created icon. This is to designate that the program version was not applied from a template but user created. Even though this program version is user created, template rules will still apply. Template created program versions automatically have variables, algorithms, and driver assignments locked. These items cannot be edited or deleted. You can add items and create additional versions. User created program versions that were built by you are open for editing, deletion or moving, unless locked.

Other domain types can contain both user created and template created program versions. RateManager domain type can only contain user created program versions. In template generated program versions, the name is not open for editing. All other items can be edited as needed.

Creating a Program Version

New program versions can be added at any time. There is no limit to the numbers of program versions you can enter.

1. In Programs, select the domain type, domain, project, product and folder that contains the

program where you want to create a new version.

2. Double-click the program. All versions of the program are displayed in the Program Detail area. For newly created programs, there is only one version.

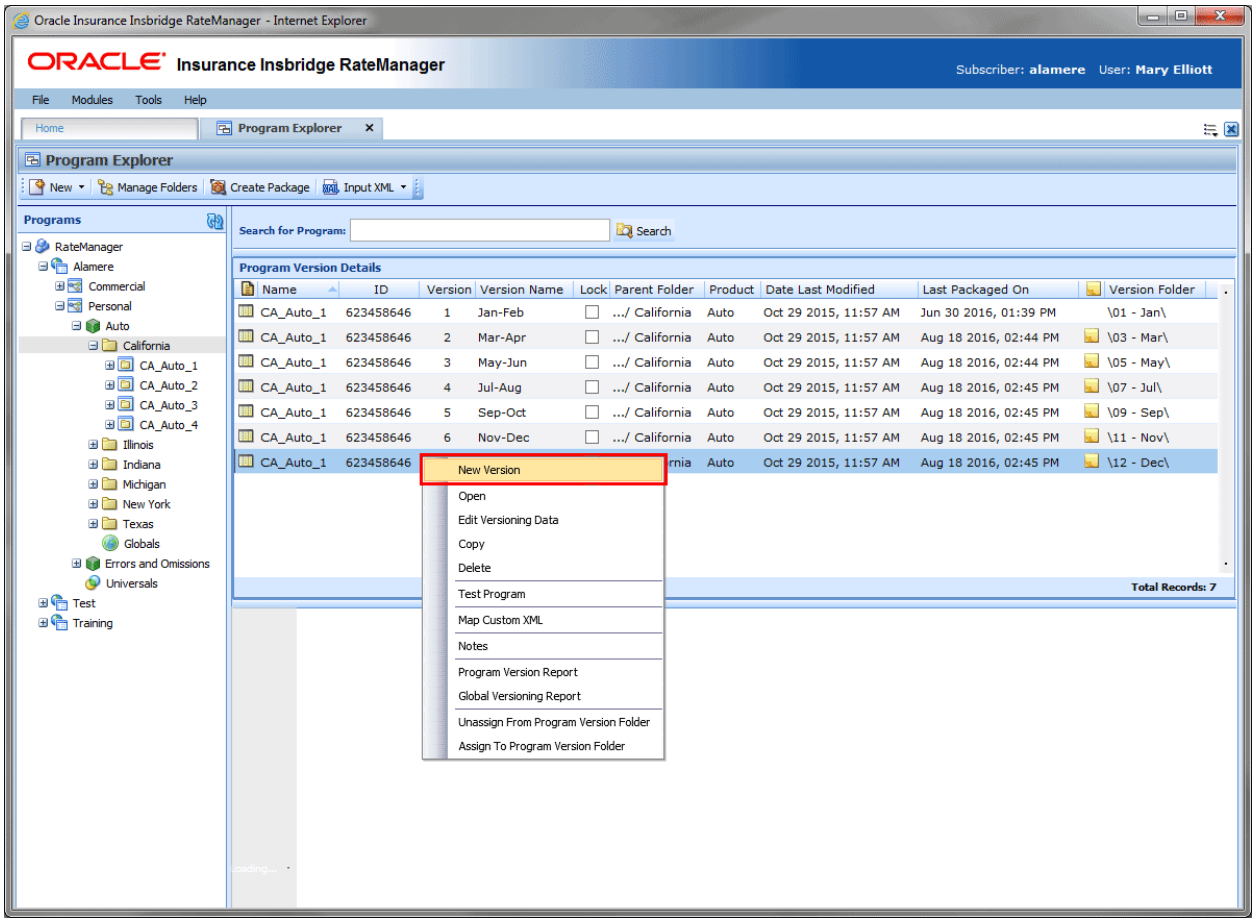


Figure 94 Adding a New Version

3. Select the version you want the new version to be based on. Right click and select **New Version**. The version is copied as the next available version number. Refresh the screen to view the new version.

NOTE: *When a new version of a program is created it is initially a pointer to all elements in the last version of the program. When a change is made to an element in that version, it can be made version specific or applied to multiple versions.*

Copying a Program Version

You can copy a program version at any time within the same project. Program versions can be copied to any folder in the project where the program is located.

1. In Programs, select the domain type, domain, project, product and folder that contains the program with the version you want to copy.

2. Double-click the program. All versions of the program are displayed in the Program Detail area. For newly created programs, there is only one version. Select the version you want to copy.

3. Right click the version and select **Copy**. This opens the Copy Program window.

4. Enter the new program name and select the folder where you want the new program copied to. You also have a choice to either copy only the definitions or the definitions and all data.

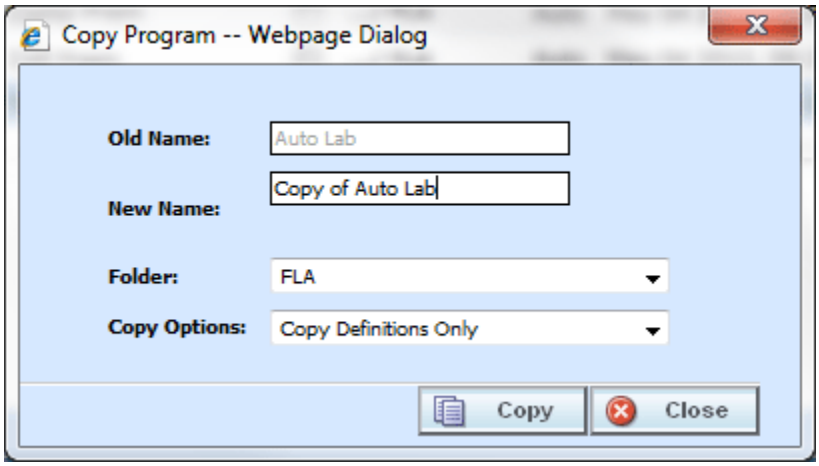


Figure 95 Copy Program Version

5. Selecting **Copy** copies the program version to the folder you selected. Selecting Close closes the window without copying the program version.

NOTE: *When you copy a program, the version description is copied also. Be sure to make any changes to the version description to ensure the correct program information is displayed on generated reports.*

Notes on Program Versions

When copying a program version or creating a new version from an existing version, a note is created on the new program version that details from where the new program version was copied or created. Any notes added to the original program version are not copied into the new program version.

To view notes, select the program version and right click. Select Notes. The notes screen is shown with the information regarding the original program.

Deleting a Program Version

If a program version is no longer needed, it can be deleted. Unless you delete the last version of a program, deleting a version does not delete algorithms, variables, or driver assignment scenarios, but it does delete the sequence and any result groups. Deleting the last version of a program is the same as deleting a program.

1. In Programs, select the domain type, domain, project, product and folder where the program version you want to delete is located.

2. Double-click the program. All versions of the program are displayed in the Program Detail area. For newly created programs, there is only one version.

3. Select the version you want to delete. Right click and select **Delete**.

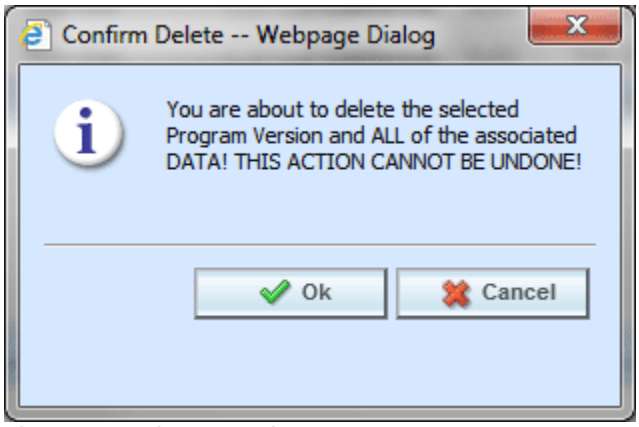


Figure 96 Delete Warning Message

4. You must confirm the deletion of the version. Select **OK** to confirm the deletion or Cancel to stop the deletion.

5. You must confirm again that you want to delete the program version. If you are deleting the last version, you will be asked to confirm deletion of the program.

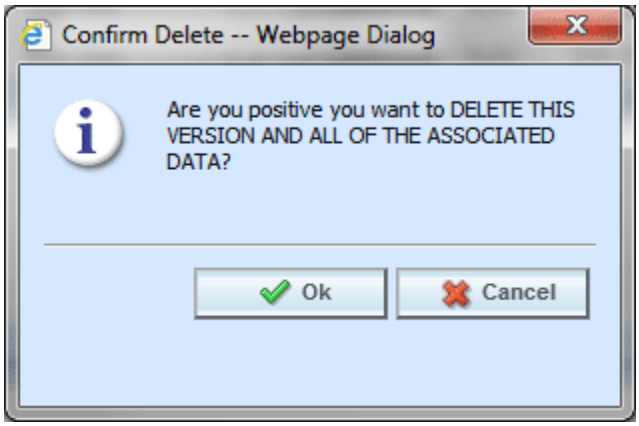


Figure 97 Confirming Delete

6. Select **OK** to delete. Click Cancel to stop the deletion.

NOTE: *Make sure this is the action you want to take. Program Version deletion cannot be undone.*

Editing a Program Version

Program versioning data and program version name can be edited at any time. Use the Edit Versioning Data option when you want to change the name of a version or enter data used for version selection.

1. Select the program version you want to edit and right click. Select **Edit Versioning Data**.

2. This opens the **Edit Program Version** screen where you can enter a program version name and update the versioning data information.

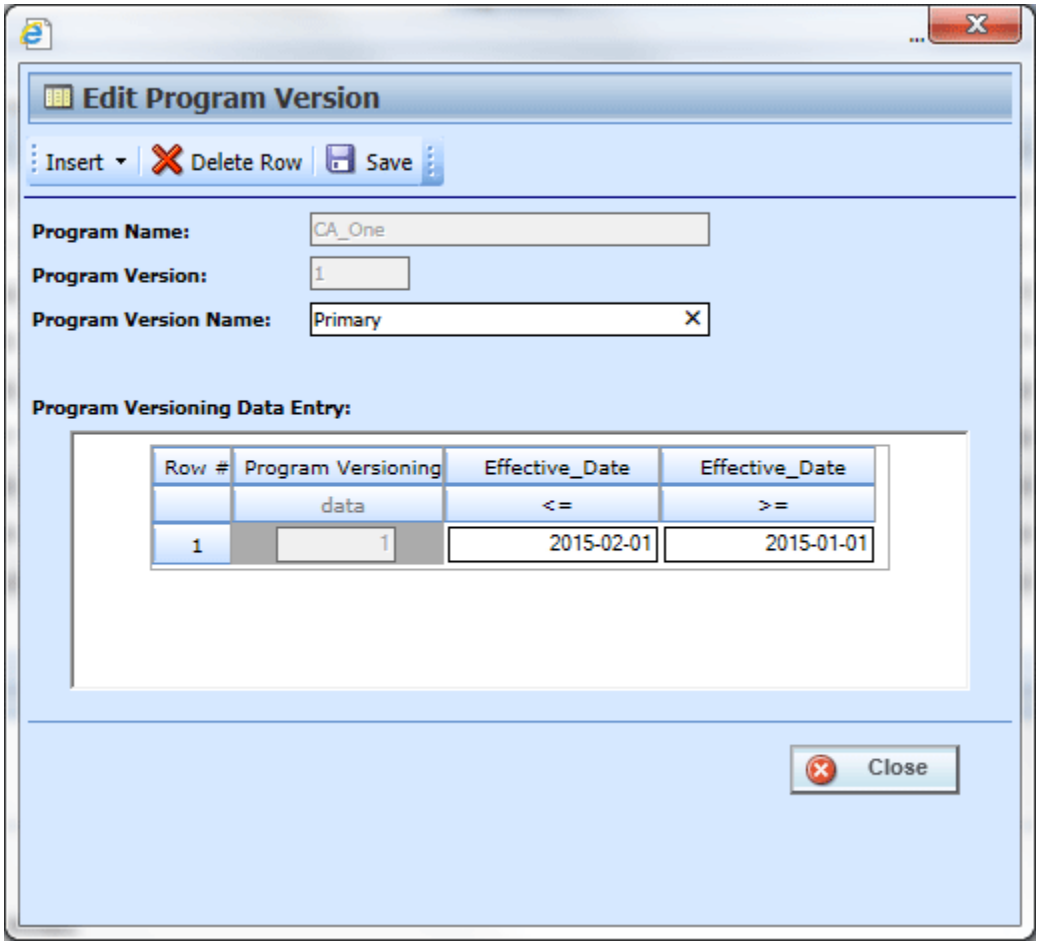


Figure 98 Editing a Program Version

3. When you have finished, click **Save** to save your changes. Click **Close** to close the Edit Program Version screen and return to Program Explorer. If you do not save prior to closing, your updates will not be saved.

VERSIONING CRITERIA

Versioning Criteria is used when you want to define which version of a program you want to use based on a policy level field. For example, if you have a program with three versions, you can select which version to use by the effective date (a policy level field). You can use version one if the effective date is this month. Version two can be used if the effective date is next month and version three can be used for any other date.

Setting up versioning criteria is a two-step process:

Step 1 – At the program level, select the criteria you want to use. This is the field that determines which version of the program should be used.

Step 2 – At the program version level, you enter the defining parameters of the field.

If you want to use this feature, you must create a global versioning package. A global versioning package allows for version criteria to be viewed at the program level. Please see Creating a Global Versioning Package for more information.

NOTE: *Only Policy level fields can be used as Version Selection Criteria.* The *maximum number of criteria that can be used is 10.*

Step 1 Select the Versioning Criteria

5. Select the program where you want to add versioning criteria and right click. Select Program Properties.
6. This opens the **Edit Program** screen where manage version selection criteria.
7. Select the criteria in the order you want them run. Or make any changes to your selections.
8. When you have finished click **Save** to save your selections or edits. Click **Close** to close the Edit Program screen and return to Program Explorer.

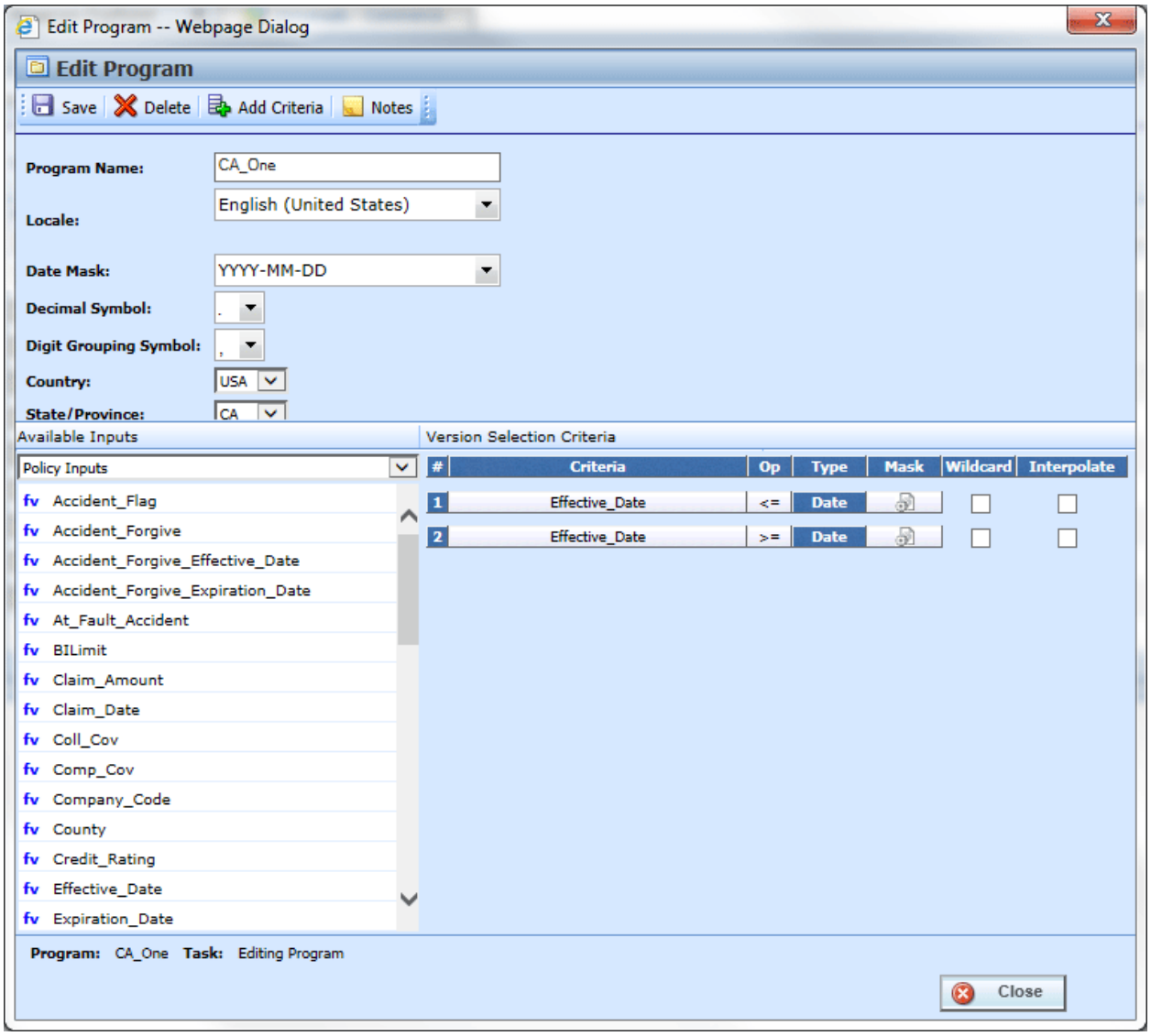


Figure 99 Adding Versioning Criteria to a Program

Versioning Criteria Functionality

The **Menu Bar** at the top of the screen contains the functions you can perform on this screen.

Save	**Save:** Saves your entries.
Delete	**Delete:** Deletes the currently selected field criteria.
Add Criteria	**Add Criteria:** Adds the selected field as criteria.
Notes	**Notes:** Enter a note. Please see Notes for more information.

Versioning Criteria

The Edit Program screen contains the program information in the top portion of the screen and the versioning criteria in the lower portion of the screen. There is also an information bar at the bottom of the screen.

Program Information:

Program Name – Name of the program. This field can be edited at any time.

Locale – The locale used in the program.

Date Mask – The date mask used by the program. For more information on date masks, please see
Date Masks.

Decimal Symbol – The decimal symbol to be used in the program.

Digit Grouping Symbol – The digit grouping symbol to be used in the program.

Country – The country where this program applies.

State/Province – The state or province where this program applies.

Versioning Criteria: Available Fields

The fields available for selection are listed. You also can create a new policy level field by scrolling to the bottom of the Available Fields list and clicking <New Field Variable>. Please see Adding a New Field from Another Area for more information.

Versioning Criteria: Version Selection Criteria

 # – Sequential order of criteria selected from **Available Fields**.

Criteria – Displays the name of each field selected as criteria for this program.

Op (operator) – Determines how the data in the table is used. The default operator is an equals sign (=). To change the operator, select the current operator, right click and choose a new operator. See Comparison Operators for more details.

NOTE: *Arithmetic comparison functions such as < and > cannot be used on string values. Only the operators of equal and not equal are available for string variables.*

Type – Shows the data type of the criteria. The type of data in the field must match that of the criteria or an error will occur.

Mask – Determines how the data being passed into the program should be interpreted by allowing you to hold, remove or replace characters being passed. (For example: If you only wanted to accept the first, second and third characters of a VIN number being passed, this

feature adds the ability to hold these characters and disregard the rest.) Click the icon to set the mask. See Masking for more information.

Wildcard – A check in this field tells the system that one or more rows will accept any value passed for these criteria. Exact matches are not required. See Wildcard for more information.

Interpolate – A check in this field instructs RateManager to use interpolation to determine the correct value for the table variable. See Interpolation for more information.

Information Bar

At the bottom of the screen is an information bar that contains the program name and the task you are performing.

Right Click Options

The Criteria and Op fields have a right click menu. To view the right click menu options, select the field you want and right click. All menu options are displayed.

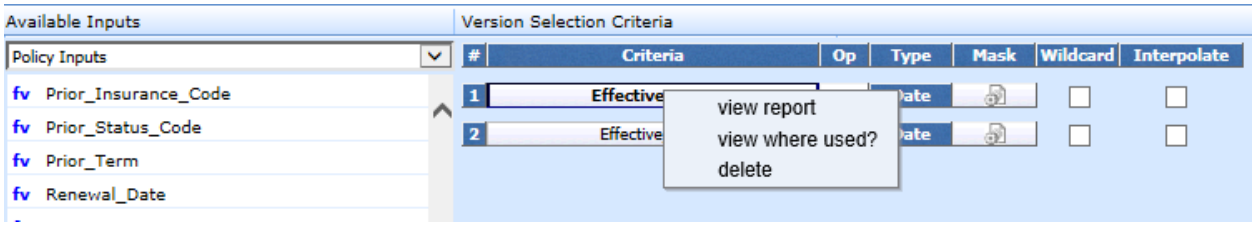


Figure 100 Right Click Menu for the Setting Versioning Criteria

The **Criteria** column has three right click menu choices:

- **View Report –** Displays field information; XML ID, Type, and Category.
- **View Where Used –** Displays the View Where Used report. Please see View Where Used for more information.
- **Delete –** Deletes the selected criteria.

The **Op** column has six right click choices:

- **Equals**
- **Not equals**
- **Less than**
- **Less than or equal to**
- **Greater than**
- **Greater than or equal to**

Setting Versioning Criteria

5. Select the field you want to use from the listing of **Available Fields**. You can either select the field and click **Add Criteria** or double-click the field. The field is displayed in the Version Selection Criteria area. A total of 10 criteria can be selected.

6. Set the comparison operator to be used when interpreting data by right clicking the operator box in the Op column (the default is an equals sign =). See Comparison Operators for more information.

7. Make your entries if you want to enter a mask, wildcard or interpolate.

8. Click **Save** to save your work. Click **Close** if you are finished entering criteria.

Deleting Versioning Criteria:

3. To remove a field as criteria, click the name in the Version Selection Criteria listing and select **Delete** from the menu bar or right click the criteria and select Delete from the menu.

4. Click **Save** to save your work. Click **Close** if you are finished editing criteria.

NOTE: *If you do not click Save your work will not be saved automatically when you exit the screen.*

Step 2 Entering Versioning Criteria

Versioning criteria are per program version.

4. Select the program version where you want to add versioning criteria and right click. Select **Edit Versioning Data**.

5. This opens the **Edit Program Version** screen where you can update the versioning data information used by the selected criteria fields. Fields cannot be added on this screen. If a field needs to be added or removed, you must return to Edit Program screen.

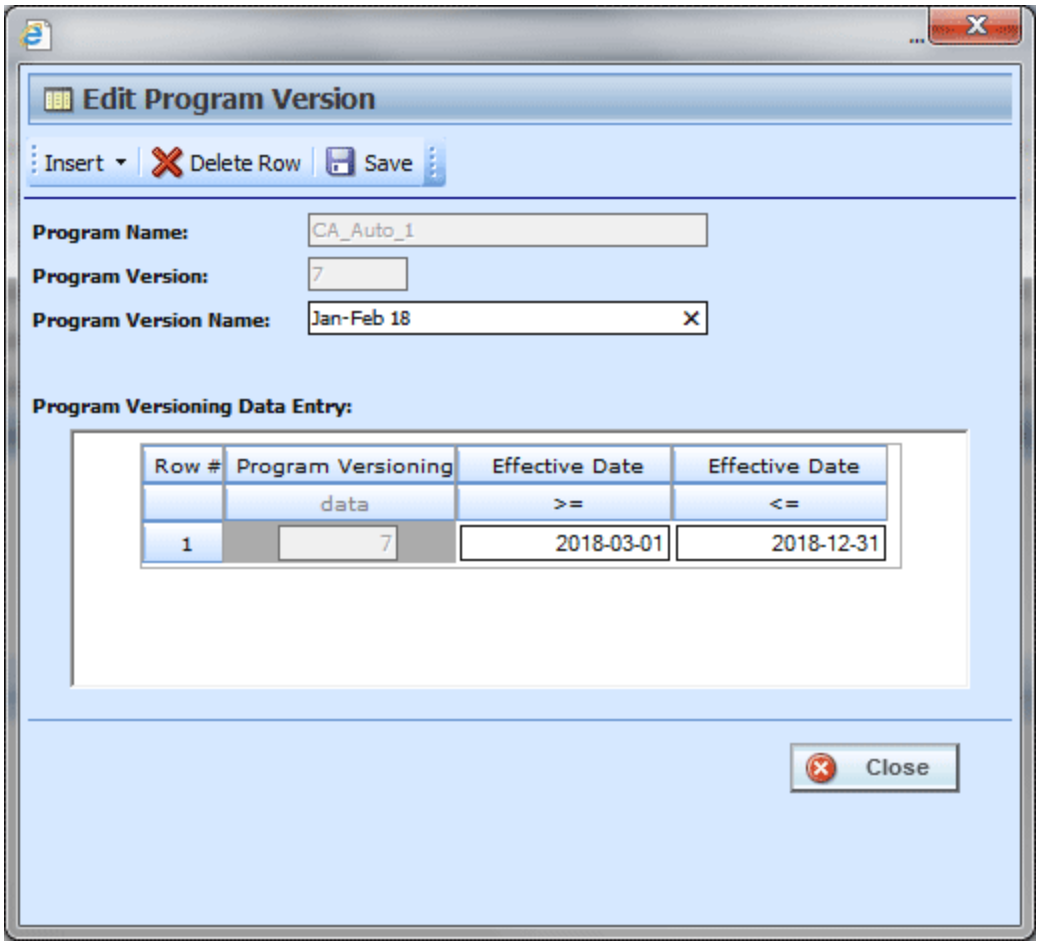


Figure 101 Editing a Program Version

6. When you have finished click **Save** to save your selections or edits. Click **Close** to close the Edit Program Version screen and return to Program Explorer. If you do not save prior to closing, your updates will not be saved.

PROGRAM VERSIONING DATA

Versioning data is used when you want to define which version of a program you want to use based on the policy level fields selected at the program level. For example, if you have a program with three versions, and you want to select which version to use by the effective date (a policy level field); use version one if the effective date is this month, use version two if the effective date is next month and version three for any other date. You would enter the program and choose the effective date from the policy fields. Then you would enter version one and set the versioning data for this month. You could do this with a less than or equal to the last day of the month. Next, enter version two and set the versioning data for next month. First set the versioning data to greater than or equal to the first of next month. Then you would add a row and set the effective date to be less than or equal to the last day of next month. Finally, leave the third version blank. This would cover any other day.

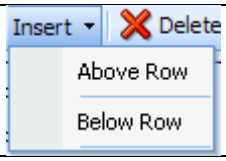
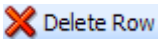
Setting up versioning data is a two-step process:

Step 1 – At the program level, select the field criteria you want to use. This is the field that determines which version of the program should be used. More than one field can be used. See Versioning Criteria.

Step 2 – At the program version level, enter the defining parameters of the field.

Program Versioning Data Functionality

The **Menu Bar** is located at the top of the screen contains the functions you can perform.

	**Insert:** Inserts a new row into the table, either above or below the currently selected row. If no row is selected, the new row is added to the end of the table.
	**Delete Row:** Deletes the currently selected row.
	Save: Saves any changes made and then closes the screen. Changes made in the table are saved immediately.

Program Versioning Data Screen Details

The Edit Program Version screen contains the program information in the top portion of the screen and the versioning data table in the lower portion of the screen.

Program Information

Program Name: The name of the program. This can be edited only from the Program Properties→Edit Program screen.

Program Version: Version number of the current version. RateManager automatically assigns this when a new version is created and cannot be edited.

Program Version Name: Provides a place for you to enter a custom name for a version.

Program version names can help you to quickly identify a specific version. Program version names can be 40 alphanumeric characters including spaces. Program version names are not required and may be edited or removed at any time. Program version names are carried through to packages and are visible in IBFA and on other areas of RateManager such as Releases.

Versioning Data Table

The data table contains the policy level fields selected at the program level. The data version parameters entered here determine if this version is to be used. When a version is being selected, the version that returns a true value based on the versioning data is the version to be used. If more than one version meets the criteria, the version with the most recent package date will be used. If no version meets the criteria, the version with the most recent package date is used.

The Program Versioning column in the data table always displays the number of the current version that is being edited. The data table can be left blank. In this case, the version with the most recent package date is the version used.

Setting Versioning Data

1. In the data table, enter the valid values that can be used to call the version of the program that you are editing.

2. Click **Close** if you are finished editing the table. Changes made to the data table are saved immediately.

Deleting Versioning Data

1. To remove a row in the data table, click the row and select **Delete Row** from the menu bar. The row will be removed.

2. Click **Close** if you are finished editing the table. Changes made to the data table are saved immediately.

MANAGING PROGRAM VERSIONS

RateManager has several features to assist with collaboration and organization of program versions.

- **Program Version Names –** Program Version Names allow you to assign meaningful names, for example effective date or program author, to program versions.
- **Program Version Folders –** Program Version Folders allow you to assign program versions to a specific folder, for example major rate changes identifiers, or SoftRater staging environments.

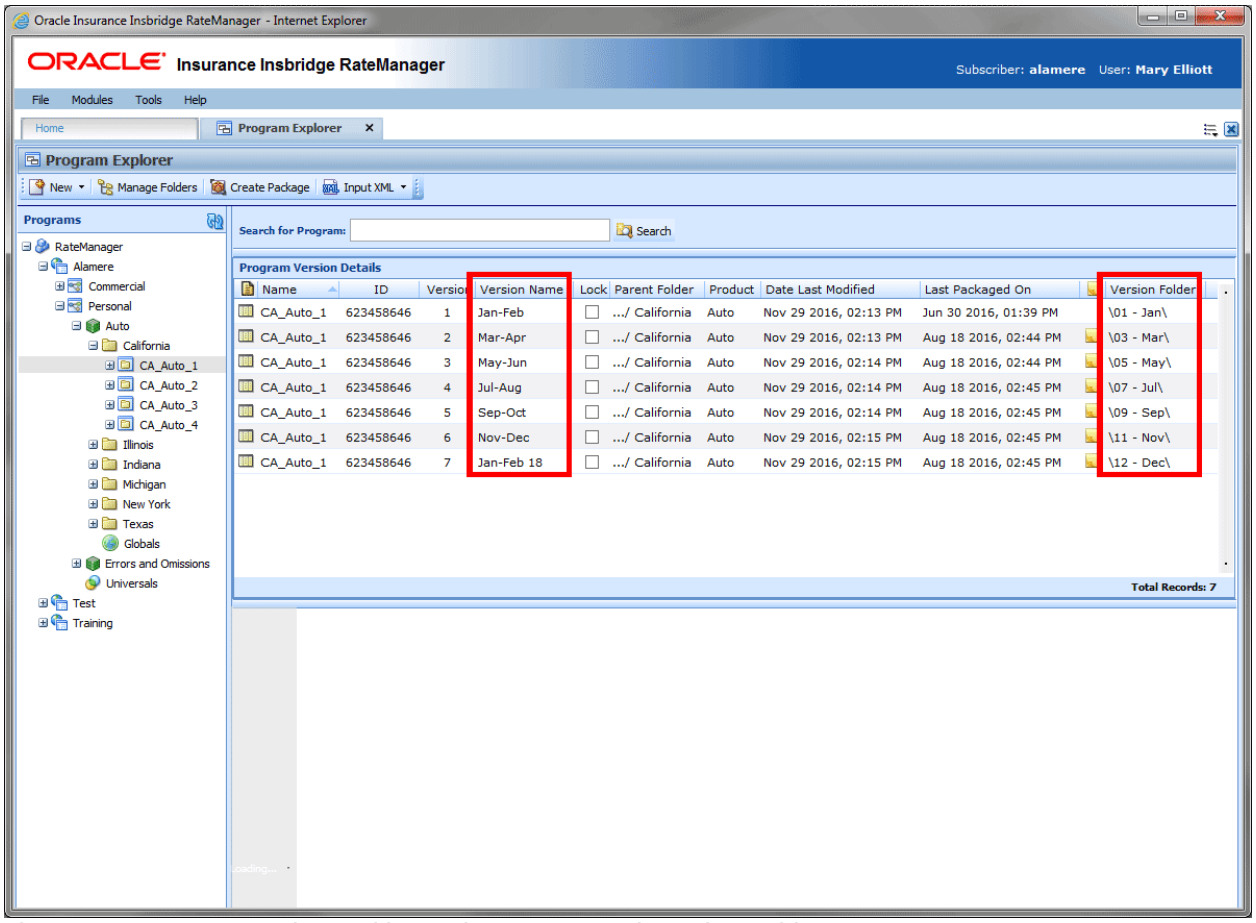


Figure 102 Program Explorer with Version Names and Version Folders

Program Version Names and Program Version Folders are displayed on the Program Explorer tab. Program Version Folders also are listed in the Programs area and in the Manage Folders screen.

Program Version Names

Program Version Names can assist with locating a specific version quickly by looking for a name instead of a program version number. Program version names can be 40 alphanumeric characters.

Program Version Names can be added to program versions at any time and do not affect the integration or the execution of SoftRater.

Program Version Names are optional. An automatically-generated version number is still created and can be used for program identification.

Entering a Program Version Name

Program Version Names are entered using the Edit Versioning option.

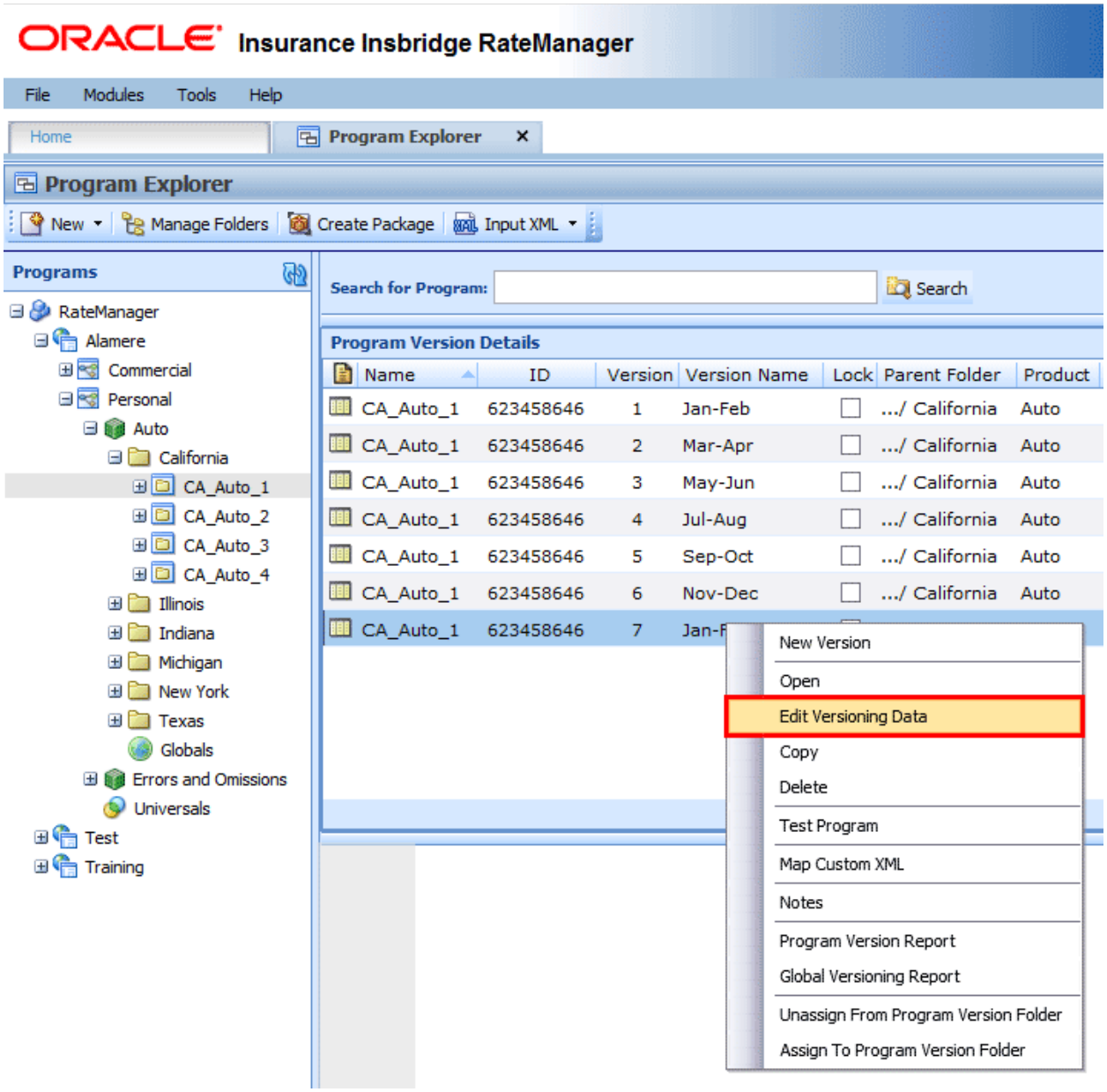


Figure 103 Creating a Program Version Name

1. To assign a name to a Program Version Name, select the program version that you want to name.

2. Right click and select **Edit Versioning Data**. The Edit Program Version screen is shown.

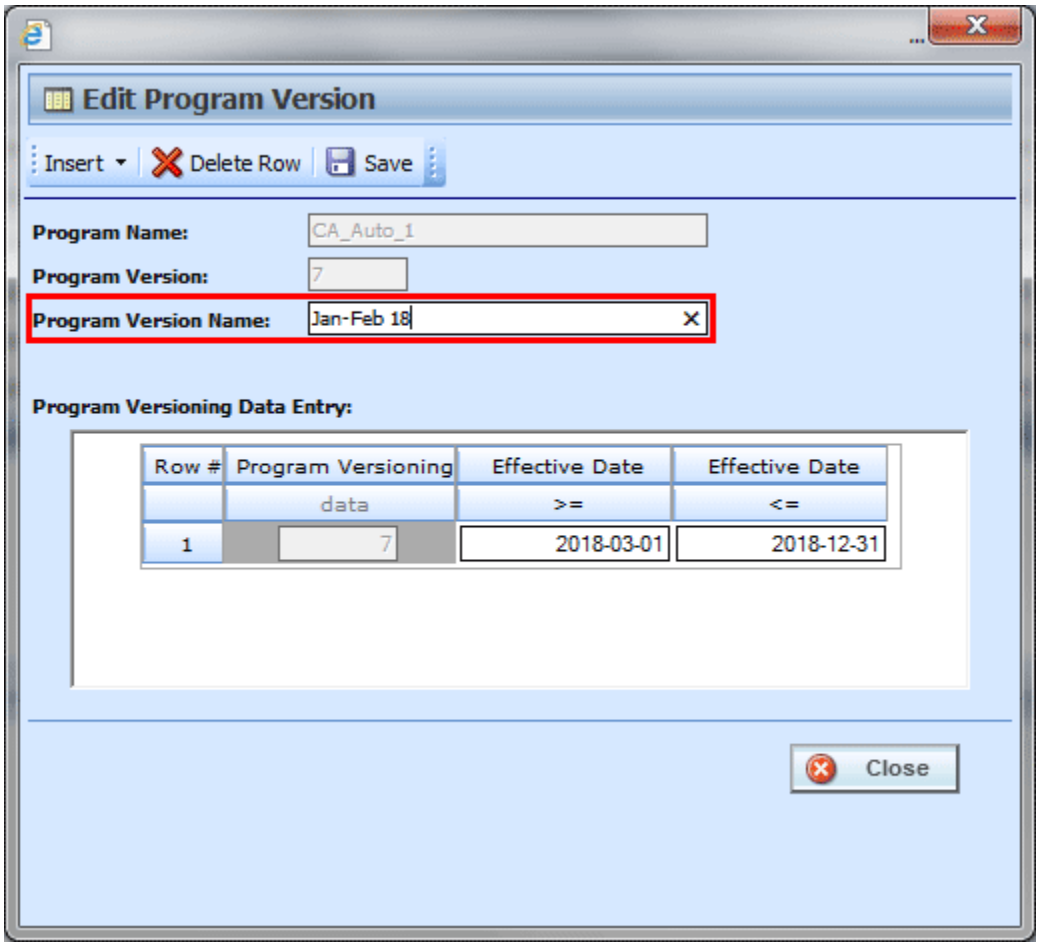


Figure 104 Entering a Program Version Name

3. Enter a **name** in the Program Version Name field.

4. Click **Save**. If you do not save prior to closing, your updates will not be saved.

The Edit Program Version screen closes. The program name is displayed in the Version Name column in the Program Detail area.

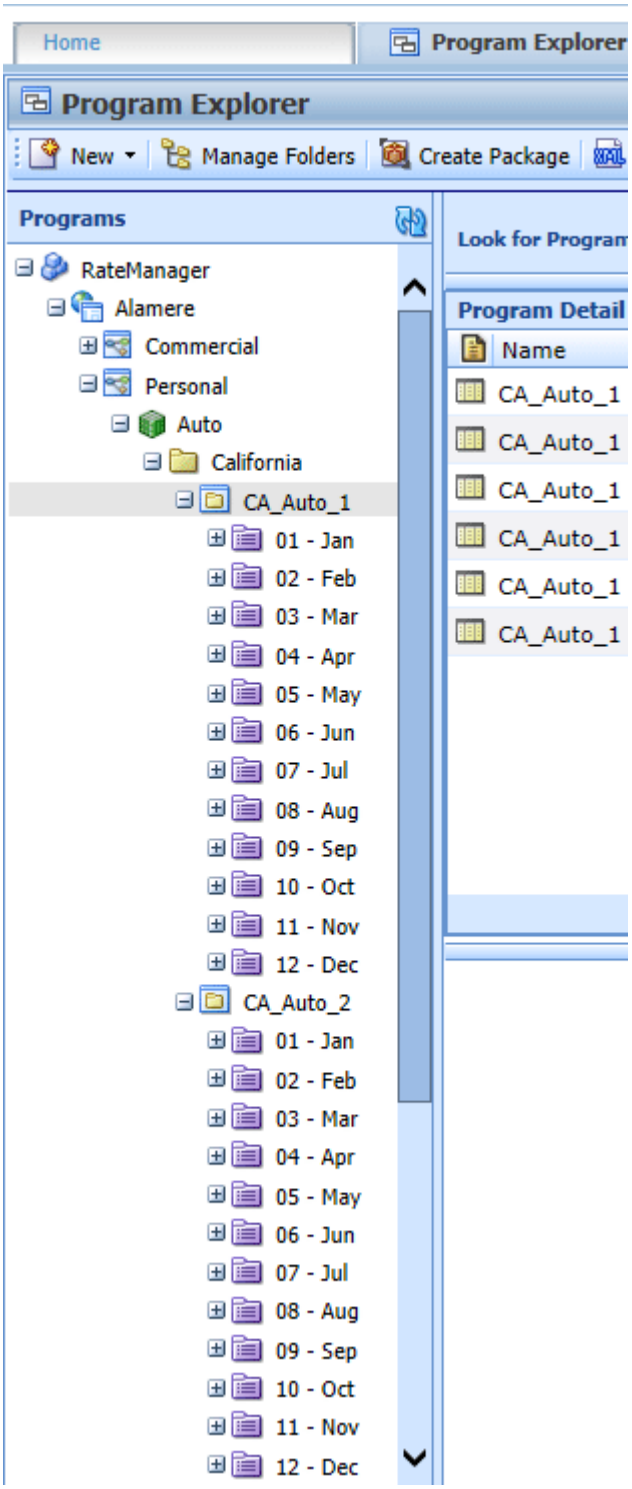
Use this method to remove or edit a program version name.

Program Version Folders

Program Version Folders are optional holding folders that can help to organize multiple program versions into manageable groupings. This may assist when you have a large number of versions that you may not need any more but do not want to delete. Or when you are updating a program version and do not want other users to access it yet.

Program Version Folders are created in the Managed Folder screen. Assigning and unassigning program versions are performed on the Program Explorer tab. For more on creating program version folders, please see

Creating a Program Version Folder.



Program Version Folder Structure:

Program version folders are nested under programs and are visible only when a program is populated in the folder. If no programs are in the folder, the program version folders are not visible.

The program version folder structure is duplicated in each program under the folder. While the folder structure is duplicated, the contents are not. Placing a program version in one folder does not also place that program version in another program version folder.

Functionality:

You can assign any program version to a program version folder located under that program. You cannot assign a program version to a program version folder under another program. Meaning all program versions found under program A can be assigned to program version folders under program A only. You cannot assign a program version found under program A to a program version folder under program B.

Program versions can be assigned to one folder only. Program versions cannot be moved but can be assigned and then unassigned multiple times.

If you want to move the program version to another folder, you can go in an assign the program version again. You do not have to unassign it. The last assignment is where the program version will be located.

Program versions can be assigned and unassigned to program version folders at any time, and do not affect the integration or the execution of SoftRater.

Program version folders act a classification method. Assigning or unassigning a program version to a folder does not remove or replace the program version from RateManager.

Deleting a program version folder does not remove the program version from

RateManager. Program versions will be disassociated from any folder but not deleted.

Assigning a Program Version Folder

Program versions can be assigned to a Program Version Folder at any time and do not affect functionality of the program version. Program versions do not need to be packaged. Locked program versions can be assigned to a Program Version Folder.

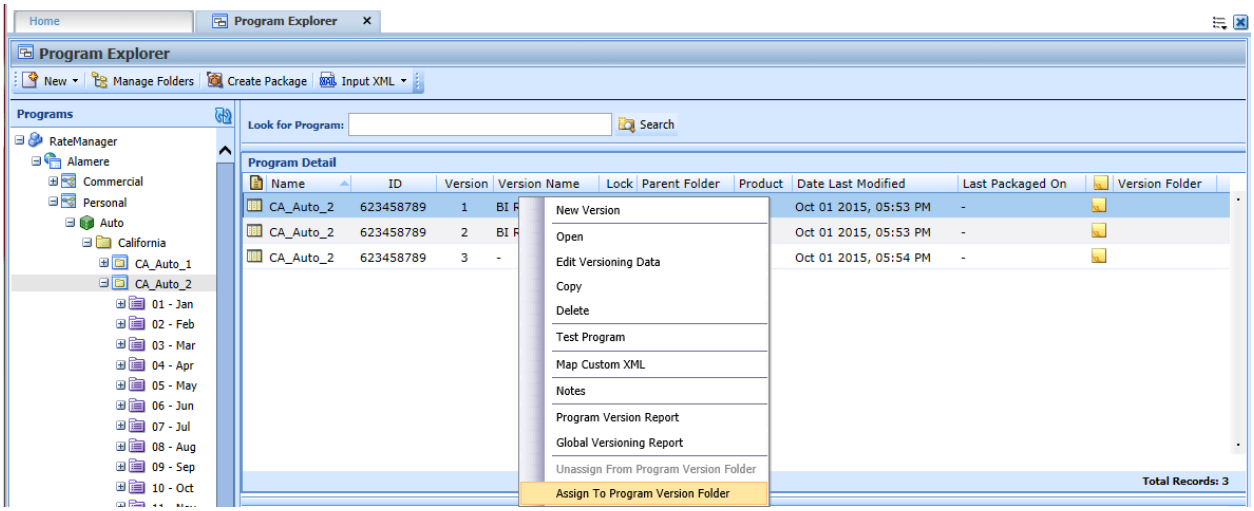


Figure 105 Assigning a Program Version Folder

1. To assign a program version to a program version folder, right-click the program version you want to assign in the Program Explorer tab.

2. Select **Assign to Program Version Folder**. The Assign Version to Program Version Folder is shown.

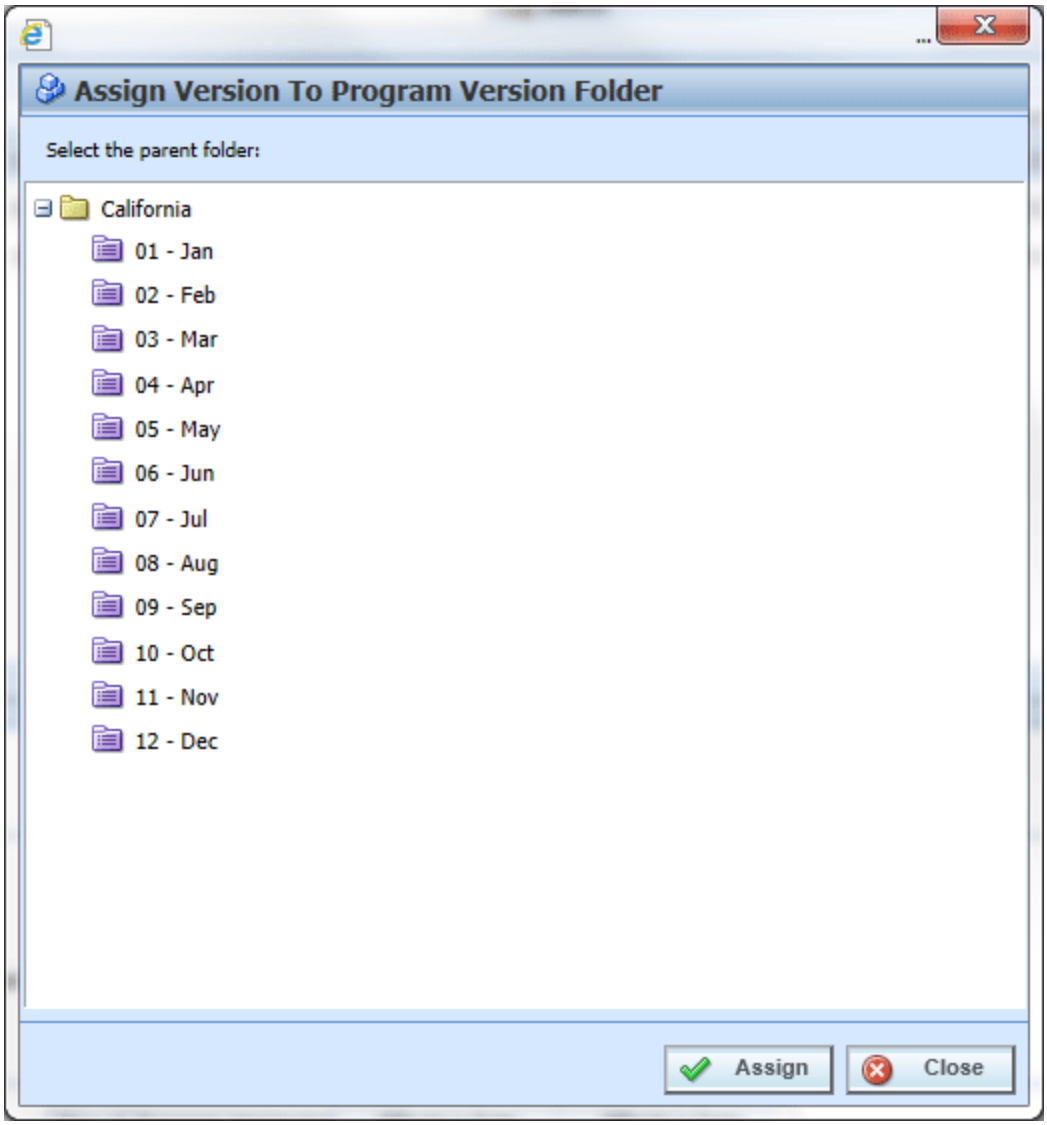


Figure 106 Assigning a Program to a Program Version Folder

3. Select the desired program version folder from the listing.

4. Click **Assign**.

The Assign to Program Version Folder screen closes and the selected program version folder is displayed in the Version Folder column in the Program Detail area.

Viewing Contents

To view the contents of a program version folder, click the folder. The versions contained are displayed in the Program Detail area.

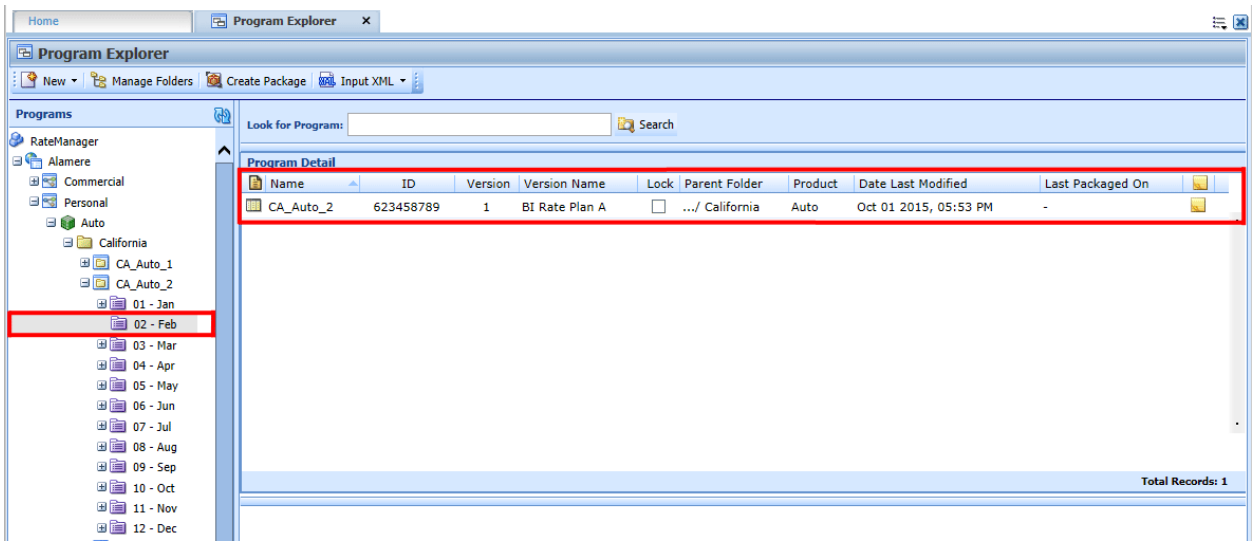


Figure 107 Viewing Program Version Folders Contents

Unassigning a Program Version Folder

Program versions can be unassigned at any time. Unassigning does not remove or delete the program version.

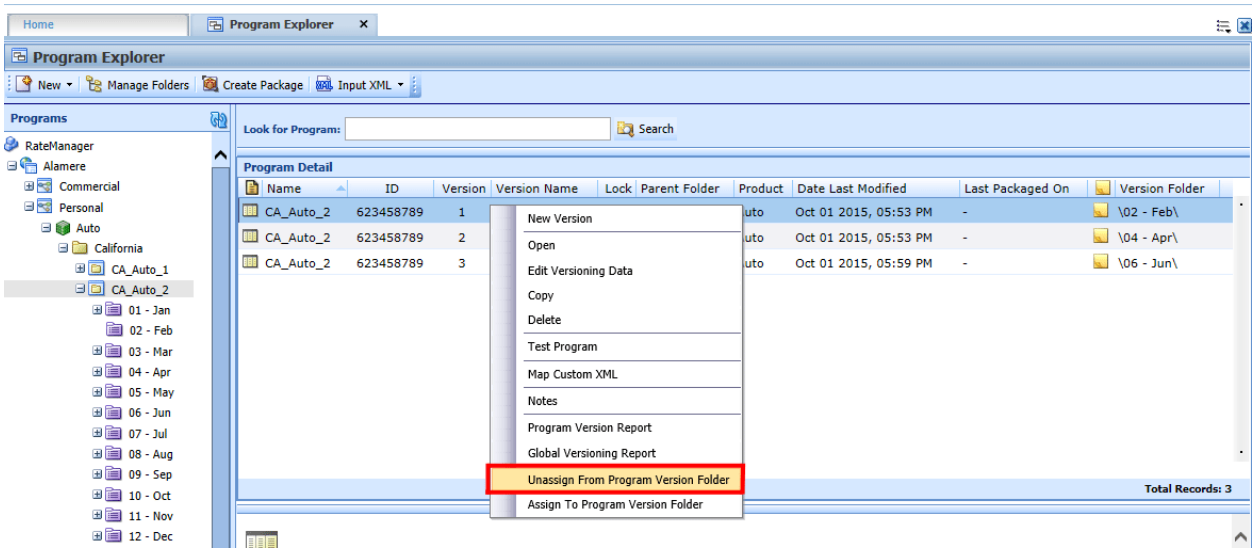


Figure 108 Unassigning a Program Version

1. To unassign a program version, select the program version you want to unassign on the Program Explorer tab.

2. Right-click and select **Unassign From Program Version Folder**.

3. A warning message is shown displayed. Select **OK** to unassign a program version. Click Cancel to keep the program version in the assigned folder.

SoftRater Automatic Version Selection

SoftRater can automatically determine what program version to use, based on the versioning criteria that were selected in RateManager. Your input XML should not specify a version, unless you want to override automatic versioning. For example, the input file may resemble this:

```
<rate project_id="99" policyId="10418124">
   <heading>
     <program parent_id="100" program_id="28">
     <c i="0" desc="Policy">
        <m i="119" n="EffectiveDate" v="02012018"/>
     </c>
     </program>
     <program parent_id="100" program_id="29">
     <c i="0" desc="Policy">
        <m i="119" n="EffectiveDate" v="05012018"/>
     </c>
     </program>
     <program parent_id="200" program_id="29" program_ver="3">
     <c i="0" desc="Policy">
        <m i="119" n="EffectiveDate" v="03012018"/>
     </c>
     </program>
   </heading>
     <c i="1" desc="Vehicle">
        <m i="386" n="VehicleId" v="1"/>
        <m i="182" n="Model Year" v="2012"/>
     </c>
     <c i="2" desc="Driver">
        <m i="367" n="DriverNum" v="1"/>
        <m i="37" n="Violations" v="0"/>
     </c>
     <c i="2" desc="Driver">
        <m i="367" n="DriverNum" v="2"/>
        <m i="37" n="Violations" v="0"/>
     </c>
</rate>
```

The above input XML calls 3 programs:

- The 1st program call is parent_id = 100, program_id = 28, with no specific version. When a version is not specified, automatic version selection is triggered. In this case, there is logic in Program 28 that tells SoftRater the correct version based on the EffectiveDate.
- The 2nd program is identical in nature to the 1st, but it calls program_id 29 instead.
- The 3rd program call is parent_id = 200, program_id = 29, and program_ver = 3. In this case, automatic versioning is overridden with a specific version call.

To enable SoftRater to automatically select a version of a program to rate, follow these steps:

1. Select which Policy fields are to be used to select a program version: see Editing a Program.
2. Specify the field values to be used by each program version: see Editing a Program Version.
3. Create a Global Versioning Package.
4. Rate an input file without specifying a program version.

PROGRAM DETAILS

Program Details serves as a central location for viewing and managing program elements. Program element folders are displayed on the left side of the tab. A listing of the elements contained in the folders is displayed in the right side of the tab.

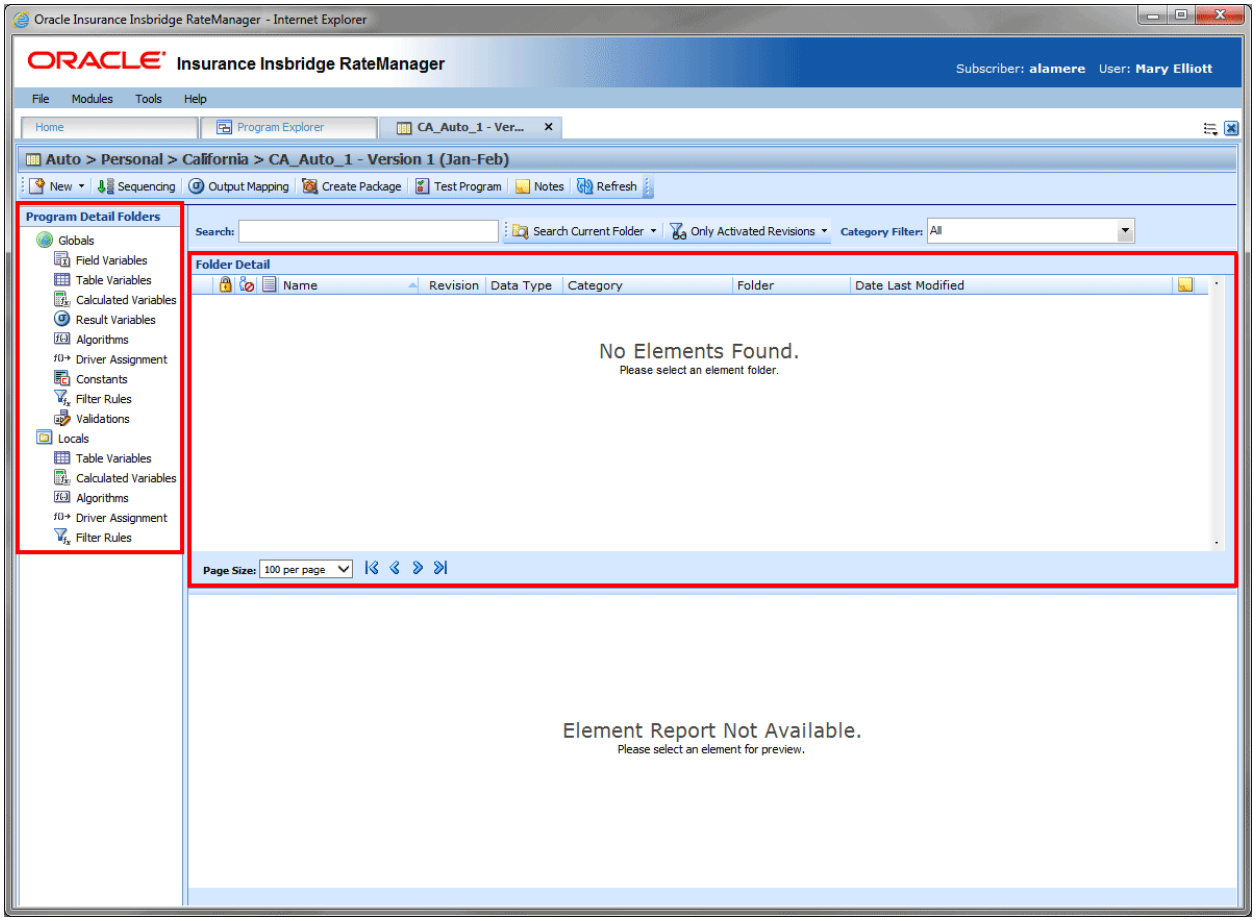


Figure 109 Program Details

Program Details is divided into two areas:

- **Program Detail Folders –** Program Details Folders is a navigation column on the left side of the tab. Global elements are displayed on top. Local elements are on the bottom. These are fixed elements. They do not expand or have right click menus. You can double click the elements to view details.

- **Folder Detail –** Folder detail lists the individual elements that you have selected from Program Details Folders. The item you select in the Program Detail Folders area determines what information is displayed in the Folder Detail area.

Navigating to Program Details

Program Details is where you manage and view the elements of a selected program version. Program Details is not an option on a menu. You must select the program version where you want to view the element details and then open it to access program details.

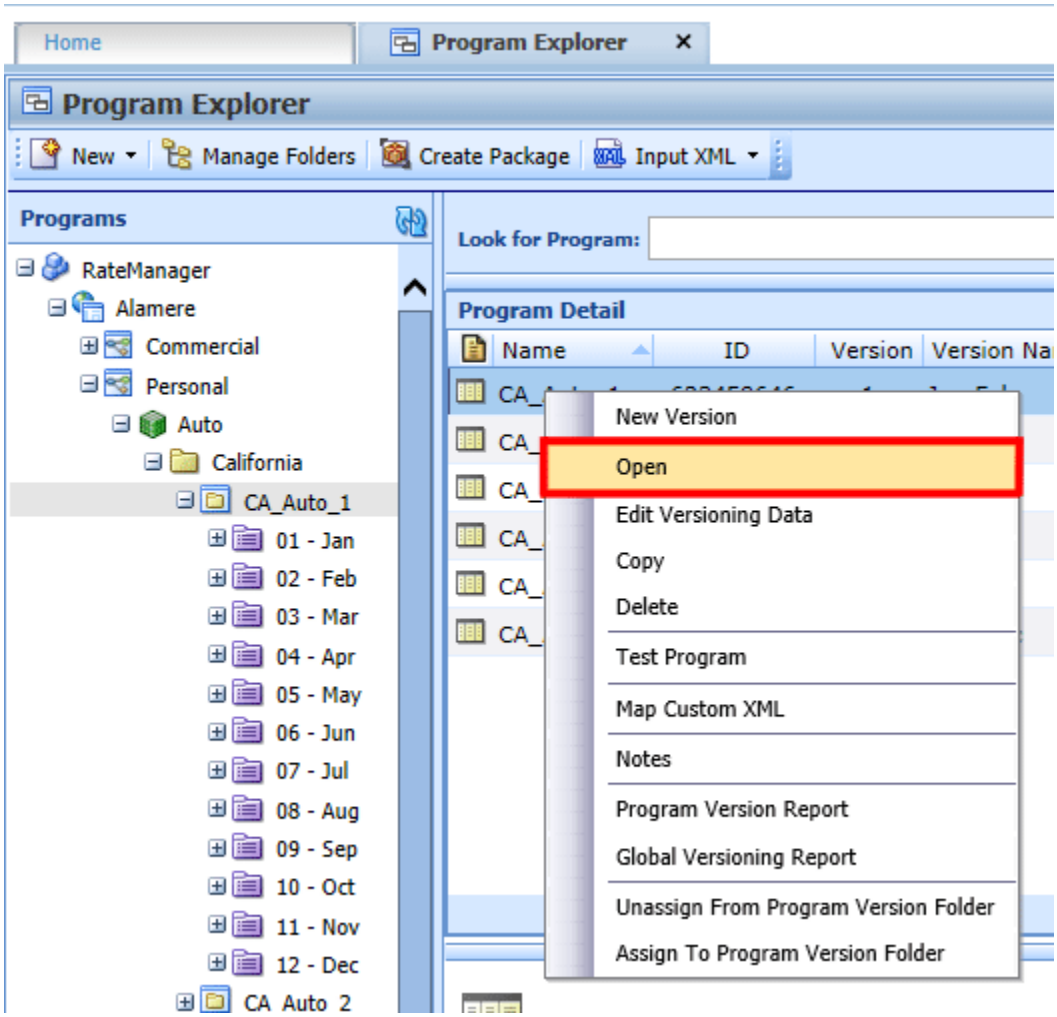
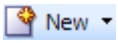
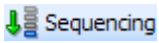
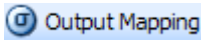
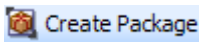
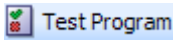
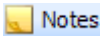
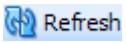


Figure 110 Navigating to Program Details

- You get to Program Details by right clicking on a program version in Program Explorer and selecting **Open**. Program Explorer is located in Modules→ Program Explorer.
- Program Details also opens when you select a recently opened program or a recently packaged program from a snap on your Home tab.

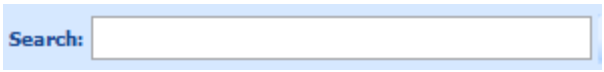
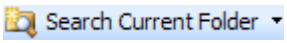
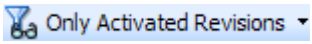
PROGRAM DETAILS FUNCTIONALITY

The **Menu Bar** is located at the top of the tab and contains the functions you can perform on this tab as well as a search option. The top bar menu has the options for creating packages, testing, and program version specific elements.

	**New:** Begins the process of creating a new element. You must select: • **Globals:** Creates a new global element. • **Locals:** Creates a new local element. After you have selected Global or Local, you can select the element you want to create. Result Variables, Calculated Variables, Constants, and Filter Rules can select **New** from the element and open the corresponding tab or window without having to select local or global from the menu.
	**Sequencing:** Opens the Sequencing tab for this program version.
	**Output Mapping:** Opens the Output Mapping tab for this program version.
	**Create Package:** Opens the Create Package function for the selected program version.
	**Test Program**: Opens the Test Program tab. Test Program lets you create, execute, debug and maintain test scenarios within the RateManager application.
	**Notes:** Opens the Notes screen for the selected element. If you want to enter a note at the version level, you must do that from Program Explorer.
	**Refresh:** Refreshes the screen with the current information.

Search

Search allows you to search for a particular element by name. Select the element where you want to search in the Program Detail Folders area. Type or paste the element name you are looking for in the Search field. You can tab out of the field or hit enter to search immediately. You can further define your search by selecting options in the search area.

	**Search:** Enter the criteria you want to search by. Input XML IDs must include the text “ID:”. Table Variables use the Variable Request ID and must include the text “ID:”. The format is: “ID:(*number*)”.
	**Search Current Folder:** You can select if you want to search only the currently selected folder or all folders.
	**Only Activated Revisions:** You can select to search only activated revisions, all revisions or all revisions that are used. This option is not available for Fields or Result Variables.
	Category Filter: You can select to search a specific category or all categories.

PROGRAM DETAIL FOLDERS

Program Detail Folders is a navigation column on the left side of the tab. Global element are displayed on top. Local elements are on the bottom. These are fixed elements. They do not expand or have right click menus. You can double click the element to view details.

PROGRAM DETAIL FOLDERS SCREEN DETAILS

	Global: Indicates that the elements underneath are the global elements of the product.
	Locals: Indicates that the elements underneath are the local elements of the program.

Right Click Menu Options

No right click Menus.

FOLDER DETAIL

The Folder Detail area lists the elements you have selected from Program Detail Folders and displays the details of any single element selected. The item you select in the Program Detail Folders area determines what information is displayed in the Folder Detail area.

There are two areas in the Folder Detail:

Element Listing – The upper portion is the element listing area. This area displays a list of the elements selected from the Program Detail Folders. For example, if you select Table Variables in the Program Detail Folders area, a list of table variables is shown in the Folder Details area.

Element Details – The lower portion of the screen contains the details of the element selected in the Element Listing area. For example, if you select a table variable, the details of the table variable is shown in the Element Details area.

FOLDER DETAIL SCREEN DETAILS

The screen details in the Element Listing area are the same for universal, global and local elements. Fields can be managed at the universal level only. Result variables can be managed at the global or universal level.

The Element Details area does not have any entries, only the details of the selected element.

NOTE: *Driver Assignment is available in the Auto product only. No other product has this option.*

Element Listing Screen Details

SELECTION	DISPLAYS
Fields	**Icon:** There are three icons: Locked. A lock in this part of the screen indicates that the field is locked. In-Use. Indicates that the field is being worked on by another user. Field. This icon indicates a field. This icon is altered when a validation has been added or if the field is deprecated. **Name:** The name of the field. **Data Type:** The type of field, i.e. date, decimal, integer or string. **Category:** The category the field is in. **Folder:** The folder the field is located in. **Date Last Modified:** The date and time the field was last updated. **Note:** A note icon indicates a note for this field. **XML ID:** The XML ID that is used to call the field.
Table Variables	**Icon:** There are six icons: Universal. This element was created at the universal level. Locked. A lock in this field indicates that the variable is locked. In-Use. Indicates that the variable is being worked on by another user. Advanced Option. Indicates that this variable contains an advanced option. Filter Rule. A filter rule has been applied. Not available to data sources. Table Variable. Indicates a table variable. OR Data Source. Indicates a data source. **Name:** The name of the variable. **Revision:** The revision that is used by the current program version. **Element Type:** For table variables, the number in parenthesis indicates the total number of variables, including this one that are linked together. **Data Type:** The type of variable, i.e. date, decimal, integer or string. **Category:** The working category of the variable. **Folder:** The folder the variable is located in.

	Date Last Modified: Time stamp of the last time the variable was saved. In the case of table variables, editing the data table will also update the time stamp. **Note:** A note icon indicates a note for this variable.
Calculated Variables	**Icon:** There are five icons: Universal. This element was created at the universal level. Locked. A lock in this field indicates that the variable is locked. In-Use. Indicates that the variable is being worked on by another user. Filter Rule. A filter rule has been applied to this variable. Calculated Variable. This icon indicates a calculated variable. **Name:** The name of the variable. **Revision:** The revision that is used by the current program version. **Data Type:** The type of variable, i.e. date, decimal, integer or string. **Category:** The working category of the variable. **Folder:** The folder the variable is located in. **Date Last Modified:** Time stamp of the last time the variable was saved. **Note:** A note icon indicates a note for this variable.
Global Result Variables	**Icon:** There are four icons: Universal. This element was created at the universal level. Locked. A lock in this field indicates that the variable is locked. In-Use. Indicates that the variable is being worked on by another user. Result Variable. This icon indicates a result variable. **Name:** The name of the variable. **Data Type:** The type of variable, i.e. date, decimal, integer or string. **Category:** The working category of the variable. **Folder:** The folder the variable is located in. **Date Last Modified:** Time stamp of the last time the variable was saved. **Note:** A note icon indicates a note for this variable.
Algorithms	**Icon:** There are six icons: Universal. This element was created at the universal level.

	Locked. A lock in this field indicates that the algorithm is locked. In-Use. Indicates that the algorithm is being worked on by another user. Advanced Option. Indicates that this algorithm contains an advanced option. Filter Rule. A filter rule has been applied to this variable. Algorithm. This icon indicates an algorithm. **Name:** The name of the algorithm. **Revision:** The revision that is used by the current program version. **Element Type:** Displays the type of algorithm. I.e. Rating or Underwriting. **Category:** The working category of the algorithm. **Folder:** The folder the algorithm is located in. **Date Last Modified:** Time stamp of the last time the algorithm was saved. **Note:** A note icon indicates a note for this algorithm.
Driver Assignments Available in the AUTO Product ONLY At the Global Level	**Icon:** There are three icons: Locked. A lock in this field indicates that the algorithm is locked. In-Use. Indicates that the algorithm is being worked on by another user. Driver Assignment. This icon indicates a driver assignment. **Name:** The name of the driver assignment. **Revision:** The revision that is used by the current program version. **Element Type:** Displays the type of driver assignment i.e. Scenario, Rank or Flag. **Category:** The working category of the driver assignment. **Folder:** The folder the driver assignment is located in. **Date Last Modified:** Time stamp of the last time the driver assignment was saved. **Note:** A note icon indicates a note for this driver assignment.
Constants	**Icon:** There are four icons: Universal. This element was created at the universal level. Locked. A lock in this field indicates that the constant is locked. In-Use. Indicates that the constant is being worked on by another user. Constant. This icon indicates a constant. **Name:** The name of the constant.

	Data Type: The type of constant, i.e. date, decimal, integer or string. **Folder:** The folder the constant is located in. **Date Last Modified:** The date and time the constant was last updated. **Note:** A note icon indicates a note for this constant.
Filter Rules	**Icon:** There are four icons: Universal. This element was created at the universal level. Locked. A lock in this field indicates that the constant is locked. In-Use. Indicates that the constant is being worked on by another user. Filter Rule. This icon indicates a filter rule. **Name:** The name of the filter rule. **Revision:** The revision that is used by the current program version. **Element Type:** Displays the type of filter rule. **Data Type:** The type of filter rule, i.e. date, decimal, integer or string. **Category:** The working category of the filter rule. **Folder:** The folder the filter rule is located in. **Date Last Modified:** The date and time the filter rule was last updated. **Note:** A note icon indicates a note for this filter rule.
Validations	**Icon:** There are six icons: Universal. This element was created at the universal level. Locked. A lock indicates that the validation is locked. In-Use. Indicates that the validation is being worked on by another user. Advanced Option. Indicates that this validation contains an advanced option. Filter Rule. A filter rule has been applied to this validation. Validation. This icon indicates a validation. **Name:** The name of the validation. **Revision:** The revision that is used by the current program version. **Element Type:** Displays the type of validation. **Data Type:** The type of validation, i.e. date, decimal, integer or string.

	Category: The working category of the validation. **Folder:** The folder the validation is located in. **Date Last Modified:** Time stamp of the last time the validation was saved. **Note:** A note icon indicates a note for this validation.
REVISIONS	**A green checkmark indicates the active revision.**
UNIVERSAL	**An overlapping circle icon indicates that this is a universal element.**

Right Click Menu Options:

The right click menu options are available on the elements displayed in the Folder Details area. The menu depends upon the element selected, if the element is locked, and the rights you have. If you do not have rights or the element is locked, the menu option is grayed out.

SELECTION	RIGHT CLICK MENU
Fields	• **Edit:** Brings up the Edit Field Variable screen where you can make edits to the category location, data type and name. • **Copy:** Creates a copy of the field. • **Delete:** Deletes the field. You must have rights to delete fields. • **Deprecate:** Deprecates the field from further use. • **View Where Used:** Displays the View Where Used report. Please see View Where Used for more information. • **View History:** Displays the Audit History of the Field. Please see View History for more information. • **Notes:** Enter, edit and view notes to the field.

Table Variables	• **Edit:** Brings up the Edit Variable tab where you can edit variable information.
	• **Copy:** Brings up the Copy Variable screen for the variable. Not available on program locked variables.
	• **Delete:** Removes the variable from the program or project. A warning message is displayed to confirm the deletion. Not available on any locked variables.
	• **Show/Hide Revisions:** Displays or hides the search results for revisions that contain the same name as this variable. This option is not available on program locked variables.
	• **Export Data:** Used for the exporting of table data. The options are:
	  • **Export Data (Tab-Delimited):** Exports table data into a tab-delimited text file. You can then make changes to the file and import the data into the same variable or a different variable. This option is for locked and unlocked table variables only.
	  • **Export Data (MS Excel Workbook):** Exports table data into a Microsoft Excel workbook where the data can be viewed. Files can be changed and brought back in to RateManager. This option has setup requirements, permissions that need to be assigned and is for locked and unlocked table variables only.
	• **Import Data:** Used for the importing of table data. The options are:
	  • **Import Data (Tab-Delimited):** Imports table data from a tab-delimited text file. Once imported, changes can be made directly in RateManager or the data can be exported out, changed, and re-imported. This option is for locked and unlocked table variables only.
	  • **Import Data (MS Excel Workbook):** Imports table data from a Microsoft Excel workbook. Once imported, changes can be made directly in RateManager or the data can be brought out, changed and brought back in to RateManager. This option has setup requirements, permissions that

	need to be assigned and is for locked and unlocked table variables only. • **View Quick Report (All Data):** Shows all table data associated with the variable. This option is for both locked and unlocked table variables. • **View Quick Report:** The variables used and criteria are displayed. This option is for both locked and unlocked table variables. • **View PDF Report:** Generates a PDF of the data table for the currently selected table variable. This option is for locked and unlocked table variables only. • **View Where Used:** Displays an information screen that details where the variable is being used. Available to all locked and unlocked variables. • **View Dependency Report:** A dependency report gives you a tree view of all the associated variables and how they relate to one another. This option is for both locked and unlocked table and calculated variables. • **View History:** Displays a history of actions taken and provides a detailed account of the action taken, who performed the action, the date the action happened, program information and any element information. Not available on locked variables. • **Table Difference Report:** Displays a report on the differences between table revisions. • **Notes:** Enter, edit and view notes to the variable.
Table Variable Revisions	**Table Variable Revisions has two additional options:** • **New Revision:** Creates a new revision of the variable. • **Set as Active Revision:** Sets the currently selected revision as the active revision. After you accept the change, the selected revision will have a green checkmark to show this is now the active revision.
Calculated Variables	• **Edit:** Brings up the Edit Variable tab where you can edit variable information. • **Copy:** Brings up the Copy Variable screen for the variable. Not available on program locked variables. • **Delete:** Removes the variable from the program or project. A warning message is displayed to confirm the deletion. Not available on any locked variables. • **Show/Hide Revisions:** Displays or hides the search results for revisions that contain the same name as this variable. Not available on program locked var. • **View Quick Report:** A report showing the breakdown of steps and the variables used. This option is for both locked and unlocked table and calculated variables. • **View Where Used:** Displays an information screen that details where the variable is being used. Available to all locked and unlocked variables. • **View Dependency Report:** Gives you a tree view of all the associated variables and how they relate to one another. This option is for both locked and unlocked table and calculated variables. • **View History:** Displays a history of actions taken and provides a detailed account of the action taken, who performed the action, the date the action happened, program information and any element information. Not available on

	locked variables. • **Notes:** Enter, edit and view notes to the variable.
Calculated Variable Revisions	**Calculated Variable Revisions has two additional options:** • **New Revision:** Creates a new revision of the variable. • **Set as Active Revision:** Sets the currently selected revision as the active revision. After you accept the change, the selected revision will have a green checkmark to show this is now the active revision.
Global Result Variables	• **Edit:** Brings up the Edit Result Variable screen where you can make edits to the category location, data type and name. • **Copy:** Creates a copy of the variable. • **Delete:** Deletes the variable. You must have rights to delete variables. • **View Where Used:** Displays the View Where Used report. Please see View Where Used for more information. • **View History:** Displays the Audit History of the variable. Please see View History for more information. • **Notes:** Enter, edit and view notes to the variable.
Algorithms	• **Edit:** Brings up the Edit Algorithm tab where you can make edits. • **Copy:** Creates a copy of the algorithm. Enter in a new name and the algorithm is copied and placed in the listing. The structure of the algorithm (including the working category and definitions) is copied. Not available on Program Locked algorithms. • **Delete:** Clicking Delete removes the algorithm from the list. A warning message is displayed to confirm the deletion. Not available on any locked algorithms. • **Show/Hide Revisions:** Displays or hides the search results for revisions that contain the same name as this variable. Not available on Program Locked algorithms. • **View Quick Report:** Clicking this option displays a report that gives a detailed view of the algorithm showing all of the steps and variables used. This screen contains the program name, revision, and working category. • **View Dependency Report:** A dependency report gives you a tree view of all the associated variables and how they relate to one another within the algorithm. • **View History:** Displays a history of actions taken and provides a detailed account of the action taken, who performed the action, the date the action happened, program information and any element information. • **Notes:** Enter, edit, and view notes to the algorithm.
Algorithm Revisions	**Algorithm Revisions has two additional options:** • **New Revision:** Creates a new revision of the algorithm. • **Set as Active Revision:** Sets the currently selected revision as the active revision. After you accept the change, the selected revision will have a green checkmark to show this is now the active revision.

Driver Assignments Available in the AUTO Product ONLY	• **Edit:** Brings up the Edit Driver Assignment tab where you can make edits. • **Copy:** Clicking Copy brings up the copy screen for the driver assignment. Enter in a new name and the driver assignment is copied and placed in the listing. The structure of the driver assignment (including the vehicle usage options/category and steps) is copied. Not available on Program Locked driver assignments. • **Delete:** Clicking Delete removes the driver assignment from the list. A warning message is displayed to confirm the deletion. Not available on any locked driver assignments. • **Show/Hide Revisions:** Displays or hides the search results for revisions that contain the same name as this variable. Not available on Program Locked driver assignments. • **View Quick Report:** Clicking this option displays a report that gives a detailed view of the driver assignment showing all of the steps and variables used. This screen contains the program name, revision, and working category. • **View Dependency Report:** A dependency report gives you a tree view of all the associated variables and how they relate to one another for this driver assignment. • **View History:** Displays a history of actions taken and provides a detailed account of the action taken, who performed the action, the date the action happened, program information and any element information. • **Notes:** Enter, edit, and view notes to the algorithm.
Driver Assignment Revisions Available in the AUTO Product ONLY	**Driver Assignment Revisions has two additional options:** • **New Revision:** Creates a new revision of the driver assignment. • **Set as Active Revision:** Sets the currently selected revision as the active revision. After you accept the change, the selected revision will have a green checkmark to show this is now the active revision.
Constants	• **Edit:** Brings up the Edit Constant screen where you can make edits to the data type and name. • **Copy:** Creates a copy of the constant. • **Delete:** Deletes the constant. You must have rights to delete constants. • **View Where Used:** Displays the View Where Used report. Please see View Where Used for more information. • **View History:** Displays the Audit History of the constant. Please see View History for more information. • **Notes:** Enter, edit and view notes to the constant.
Filter Rules	• **Edit:** Brings up the Edit Filter Rule tab where you can make edits. • **Copy:** Creates a copy of the filter rule. Enter in a new name and the filter rule is copied and placed in the listing. The structure of the filter rule (including the working category and definitions) is copied. Not available on Program Locked filter rules. • **Delete:** Clicking Delete removes the filter rule from the list. A warning message is displayed to confirm the deletion. Not available on any locked filter

	rule. • **Show/Hide Revisions:** Displays or hides the search results for revisions that contain the same name as this variable. Not available on Program Locked filter rules. • **View Quick Report:** Clicking this option displays a report that gives a detailed view of the filter rule showing all of the steps and variables used. This screen contains the program name, revision, and working category. • **View Where Used:** Displays the View Where Used report. Please see View Where Used for more information. • **View Dependency Report:** A dependency report gives you a tree view of all the associated variables and how they relate to one another within the filter rule. • **View History:** Displays a history of actions taken and provides a detailed account of the action taken, who performed the action, the date the action happened, program information and any element information. **Notes:** Enter, edit, and view notes to the filter rule.
Filter Rule Revisions	**Filter Revisions has two additional options:** • **New Revision:** Creates a new revision of the variable. • **Set as Active Revision:** Sets the currently selected revision as the active revision. After you accept the change, the selected revision will have a green checkmark to show this is now the active revision.

Validations	• **Edit:** Brings up the Edit Variable tab where you can edit variable information.
	• **Copy:** Brings up the Copy Variable screen for the variable. Not available on program locked variables.
	• **Delete:** Removes the variable from the program or project. A warning message is displayed to confirm the deletion. Not available on any locked variables.
	• **Show/Hide Revisions:** Displays or hides the search results for revisions that contain the same name as this variable. This option is not available on program locked variables.
	• **Export Data:** Used for the exporting of data. The options are:
	  • Export Data (Tab-Delimited): Exports table data into a tab-delimited text file. You can then make changes to the file and import the data into the same variable or a different variable. This option is for locked and unlocked table variables only.
	  • Export Data (MS Excel Workbook): Exports table data into a Microsoft Excel workbook where the data can be viewed. Files can be changed and brought back in to RateManager. This option has setup requirements, permissions that need to be assigned and is for locked and unlocked table variables only.
	• **Import Data:** Used for the importing of table data. The options are:
	  • Import Data (Tab-Delimited): Imports table data from a tab-delimited text file. Once imported, changes can be made directly in RateManager or the data can be exported out, changed, and re-imported. This option is for locked and unlocked table variables only.

	• Import Data (MS Excel Workbook): Imports table data from a Microsoft Excel workbook. Once imported, changes can be made directly in RateManager or the data can be brought out, changed and brought back in to RateManager. This option has setup requirements, permissions that need to be assigned and is for locked and unlocked table variables only. • **View Quick Report (All Data):** Shows all table data associated with the variable. This option is for both locked and unlocked table variables. • **View Quick Report:** The variables used and criteria are displayed. This option is for both locked and unlocked table variables. • **View PDF Report:** Generates a PDF of the data table for the currently selected table variable. This option is for locked and unlocked table variables only. • **View Where Used:** Displays an information screen that details where the variable is being used. Available to all locked and unlocked variables. • **View History:** Displays a history of actions taken and provides a detailed account of the action taken, who performed the action, the date the action happened, program information and any element information. Not available on locked variables. • **Table Difference Report:** Displays a report on the differences between revisions. • **Notes:** Enter, edit and view notes to the variable.
Validations	**Validations has one additional option:** • **Set as Active Revision:** Sets the currently selected revision as the active revision. After you accept the change, the selected revision will have a green checkmark to show this is now the active revision. Setting the active revision can only be done at the global level.

Scrolling Through Element Listing

If you have a long list of elements on the screen, select the scroll bar and drag it. An information box displays the element you are currently on. When you locate the element you need in the information box, release your cursor. If you are scrolling down, the element is displayed at the bottom of the screen. If you are scrolling up, the element is displayed at the top of the screen. The Folder Detail area may not show the element until you release your cursor.

This feature is helpful when you want to scroll through a long list of elements. Or see the last element on a long list.

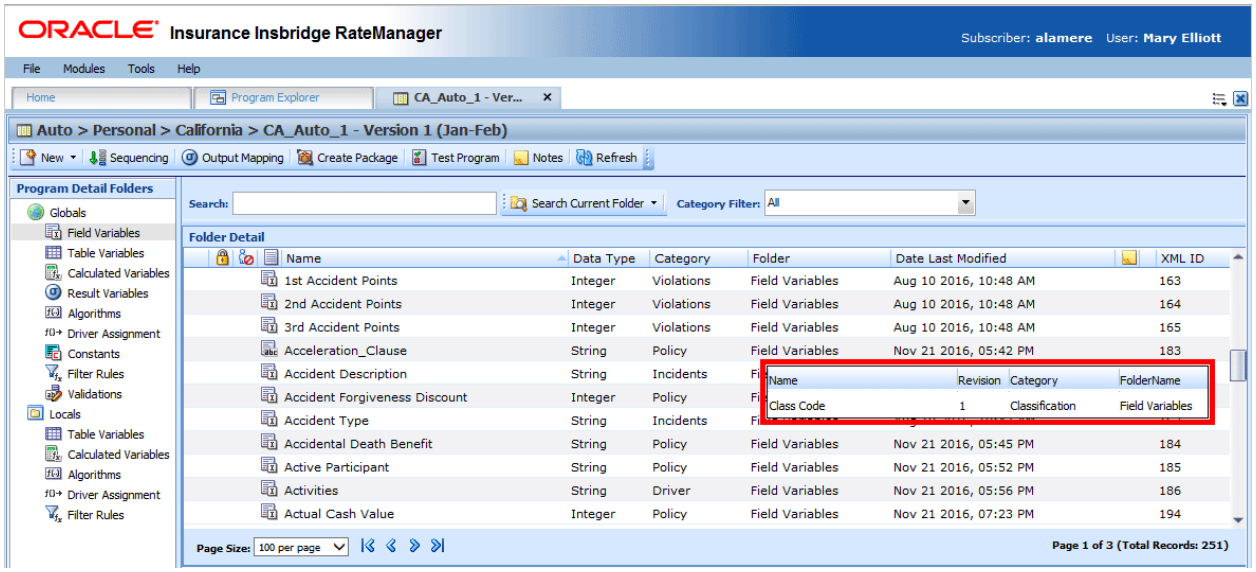


Figure 111 Element Information when Scrolling

Element Details

The element details area displays the details of the selected element. No actions can be performed in this area.

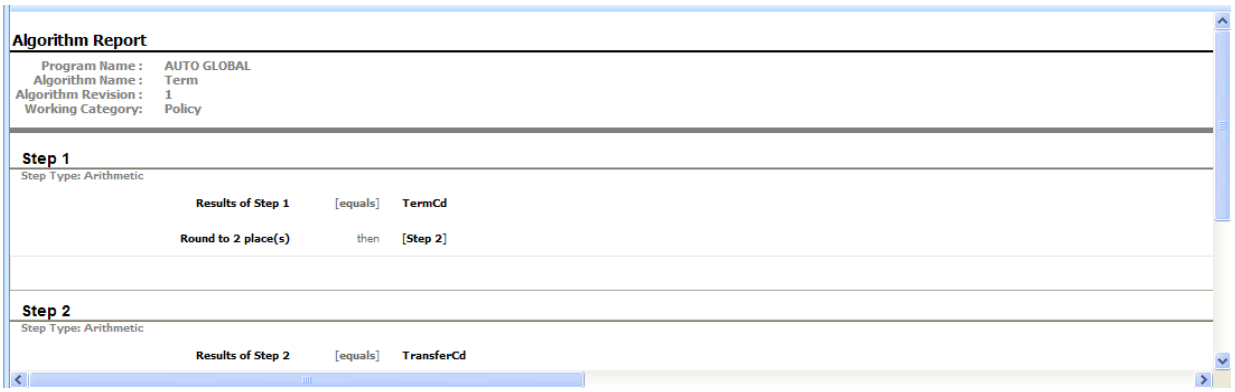
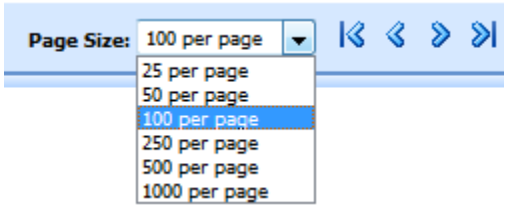


Figure 112 Element Details

Page Sizing

At the bottom of the element listing are the page size and page navigation options. The *Page Size* option is where you can select the number of elements you want displayed on the screen. The default setting is 25 per page. A total is listed to the right of the page size option. You can change the number of elements displayed at any time by selecting the page size option you want from the menu.

Page size is set per tab and applies to all elements on the Program Details tab. For example, if you select to view 100 per page on the Program Details tab then select an element folder, the page size remains 100.



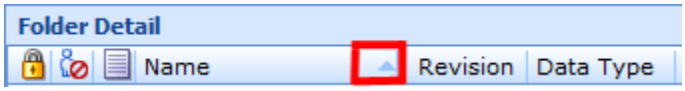
The *Page Navigation* options allow you to move back and forth between pages.

	Move to the first page
	Move back one page
	Move forward one page
	Move to the last page

Located on the far right is the total number of elements and the page you are currently on. You can place your cursor on the bottom of the element listing area and expand or contract the size of the area.

Sorting:

You can sort columns by individual column headers. The default view is alphabetically (A to Z) by Name. To sort by a different column header, click the column header you want and the current results are re-sorted. You can sort results as many times as you want.



Sorting does not filter results. It only rearranges the order in which they are displayed. This applies to all screens in Program Details. You cannot assign a sorting preference per element.

The sort icon indicates the field that is determining the sort order.

- Elements are sorted A to Z or lowest to highest.
- Elements are Z to A or highest to lowest.

Chapter 11

GLOBALS AND UNIVERSALS

When an element is created under a specific program, it is referred to as a local element. When an element is created that can be used by all programs under the product, it is referred to as a global element. When an element is created that can be used by all programs under the project, it is referred to as a universal element. If used properly, universal and global elements can greatly reduce the time taken to load and maintain the same tables or logic in multiple programs.

Table variables, calculated variables, algorithms, driver assignments and filter rules can be created at the local level and used only in the program where they were created. Result, calculated variables, table variables, algorithms, driver assignments, constants, and filter rules can be created at the global level and can be used in any program in the same product. Fields, calculated variables, result variables, table variables, algorithms, driver assignments, constants, filter rules, and validations can be created at the universal level and can be used in any program in the same project.

NOTE: *Field variables and validations are created at the universal level only.*

Globals are product specific and should be used only if the item being created is identical across all programs in which it is used. For example, if your violation code is standard and used in every program in the product, a global table variable can be created and used as an established option for all of the programs in the product. This is also optimal for maintenance.

When a global is changed, it automatically updates in every program.

Universals are project specific and should be used only if the item being created is identical across all programs in every product in which it is used. For example, if your effective date is standard and used in every program in the project, a universal field can be created. This brings consistency to every program in the project.

When a universal is changed, it automatically updates in every program.

Universal Elements

Fields, variables, algorithms, driver assignments, constants, filter rules, and validations can be created by the user in the RateManager domain type. In an Other domain type, these elements can be imported or brought in from a template. Universals that are user created or imported are open for editing, copying, and deletion if they are not locked. Universals that were brought in from a template are locked at all times and cannot be unlocked. No editing or deleting is allowed. This is to ensure the integrity of the template. New revisions can be added to a template generated element. These new revisions are open for editing, copying, or deletion, unless the program version is locked.

Working with Universal Variables

- At the universal level, packages cannot be created and program testing cannot be done.
- Only universal variables, fields or results may be used in universal table variables, algorithms, driver assignments, or filter rules. You cannot use any local or global elements.

- Universal elements are available for use by any of the programs under the project where the element is defined.
- Universal elements have the active revision set at the program version level. The default active revision is revision 1. If there is another revision that you want to use, you will have to go into the program version and set the revision there. You cannot change the active revision in the universal tab.

UNIVERSAL TAB

The universals tab is available from the Program Explorer. This tab lists only the universal elements associated with the selected project. Most of the features and functionality on this tab also can be found on Program Details with the exception of Categories. If you want to view or manage Categories, you must do so from the universals tab.

1. Select the domain type, domain, and project you need.
2. A Universal icon is listed underneath the products.

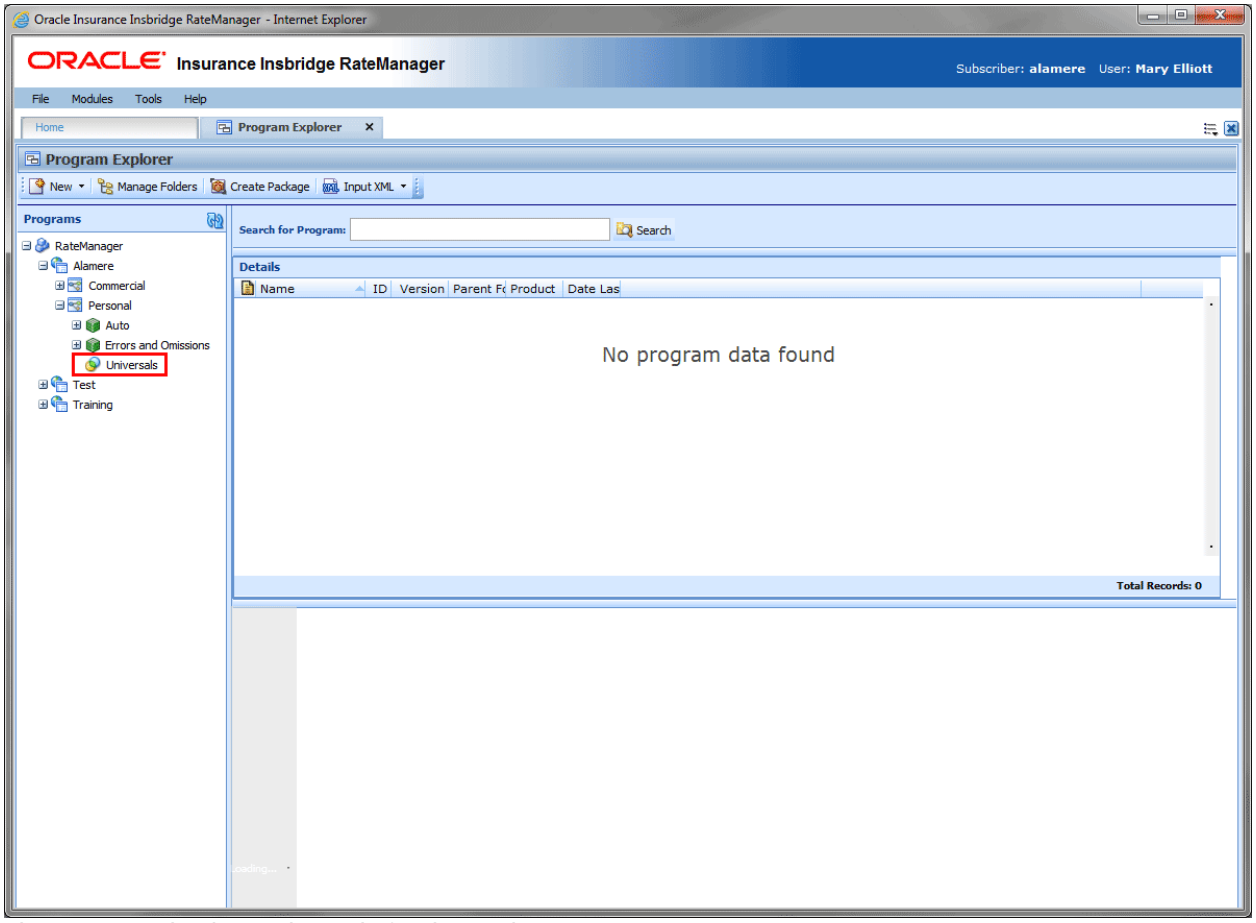


Figure 113 Selecting Universals for the Project

3. Double click the universal icon and to open the Universal Tab.

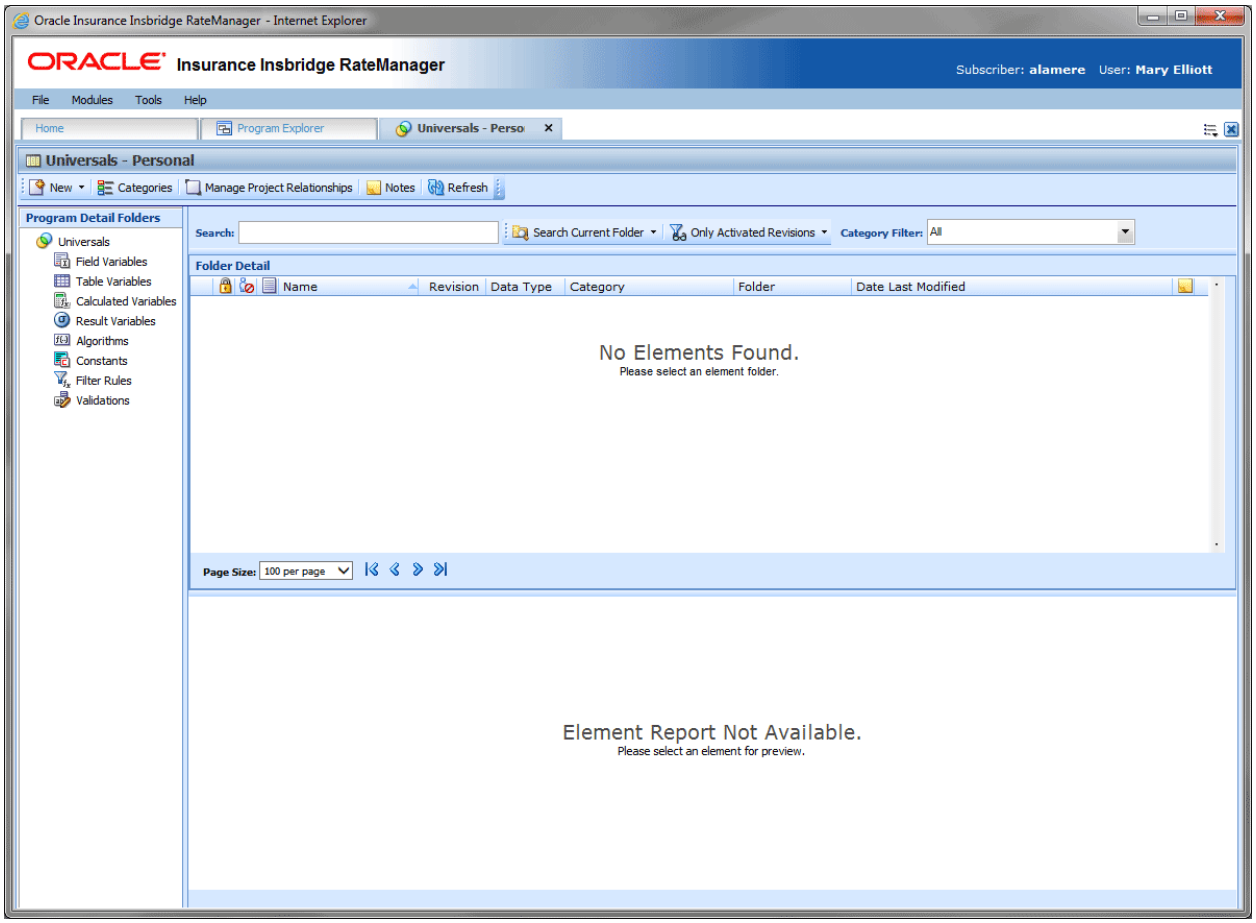
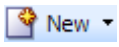
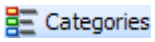
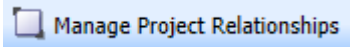
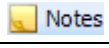
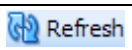


Figure 114 Universal Tab

UNIVERSAL TAB FUNCTIONS

The Universal tab functions like the Program Details tab.

	**New:** Begins the process of creating a new universal element. Fields, Result Variables, Calculated Variables, Tables Variables, Constants, Filter Rules, and Validations allow you to click New from the element and open the corresponding tab or window. You must select the type of table variable or algorithm you want to create from the menu.
	**Categories:** Opens the Category screen where you can view and manage categories.
	**Manage Project Relationships:** Opens the Project Relationships screen where you can setup relationships between projects.
	**Notes:** Opens the Notes screen for the selected element.
	**Refresh:** Refreshes the screen with the current information.

Universal Tab Screen Details

The Universal tab functions like the Program Details tab with the exception of setting revisions, managing categories and managing project relationships. Revisions cannot be set at the global level or universal level. Categories and project relationships cannot be managed elsewhere.

MANAGE PROJECT RELATIONSHIPS

Manage Project Relationships allows for callouts to be made to other programs in other projects. Each project can set up a single relationship with one other project.

For example, in the Alamere domain, the Personal project can establish a relationship with the Training domain, Project Training project. The Alamere domain cannot set up any other relationship with any other project. The Alamere domain can be edited to form a new relationship with a different project. The relationship with the Project Training project would not remain.

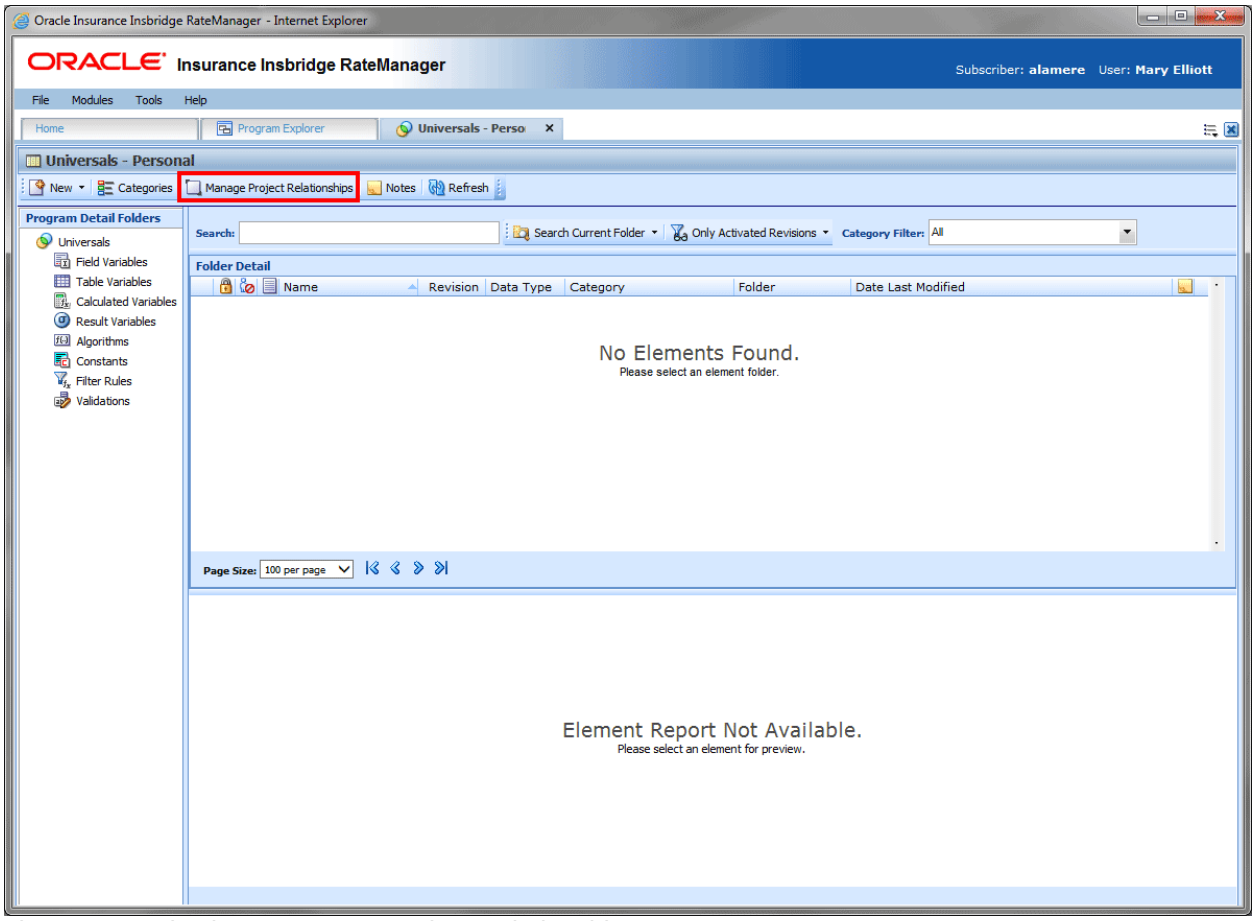


Figure 115 Selecting to Manage Project Relationships

You cannot create a relationship within the same project. If you are accessing the Manage Project Relationship screen from the Personal project, Personal will not be an option on the screen.

Relationships are granted per project. There are no default relationships. You cannot create a relationship with a project that does not contain any programs.

Relationships are needed if you want to make a callout to a program in another project. Relationships must be granted prior to creating any callouts. Until a relationship is granted, the option to callout to another project will not be available.

Navigating to Manage Project Relationships

1. Select the domain type, domain and project where you want to establish a project relationship.
2. Click the universals icon to open the universals tab.
3. Click the Manage Project Relationship button in the top bar menu.

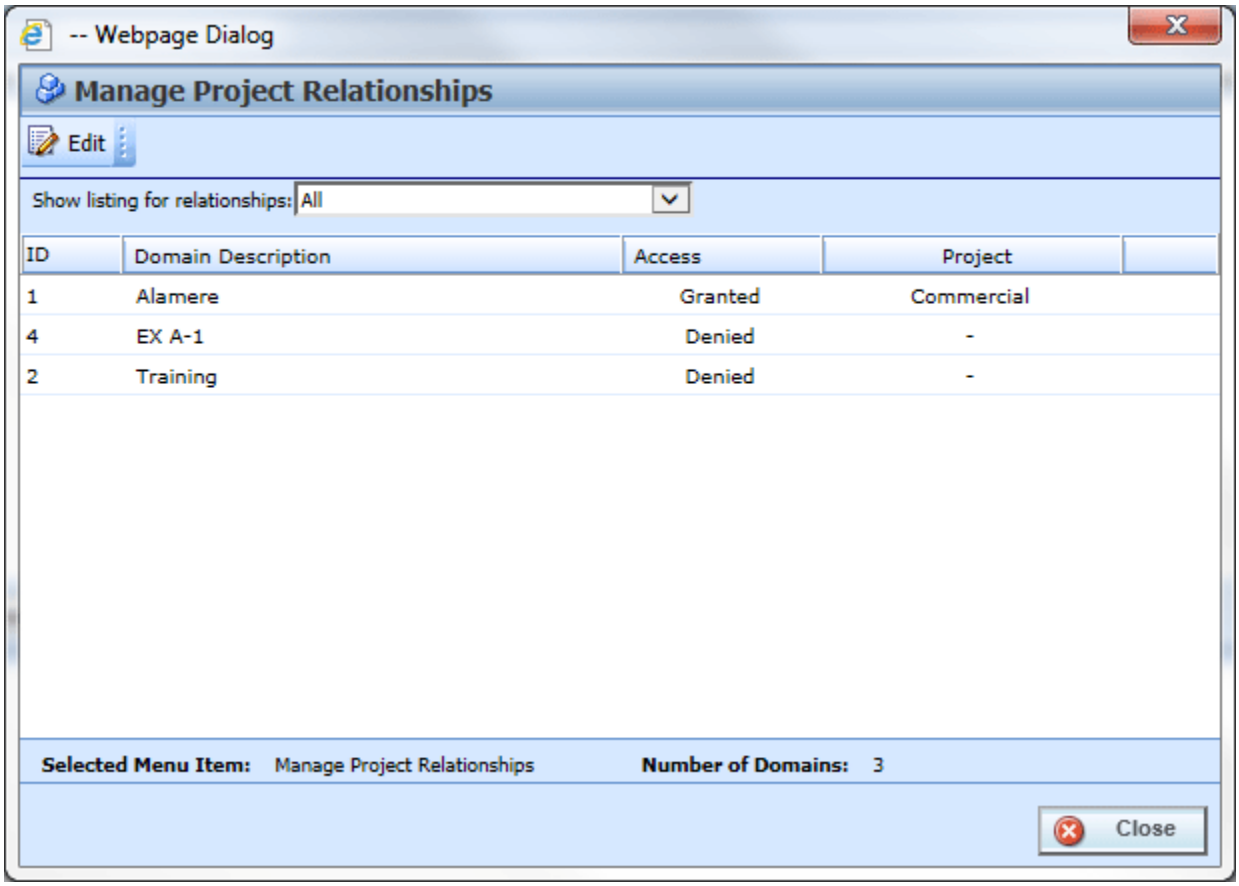


Figure 116 Editing Cross project Relationships

Functionality

	Edit: The Edit button opens the screen where you can grant access to the selected project.
	Show listing for Relationships: Filters the project by: • **All:** All projects are displayed.

	• **Activated:** Only active projects are displayed. • **Deactivated:** Only deactivated projects are displayed.

Details

ID: The XML ID for the product.

Domain Description: The Name description for the domain where the product is located.

Access: Whether access has been granted or denied

Project: The project name that this project can call out to.

Granting Access

To grant access, make sure you are in the project from where you want to callout. This is the location of the calling program.

1. On the Manage Project Relationships screen, select the listing you want to view; all projects, only activated projects or only deactivated projects. The screen refreshes with your selection.
2. Select the domain you where you want to create a relationship. This holds the target project.
3. Click the Edit button.

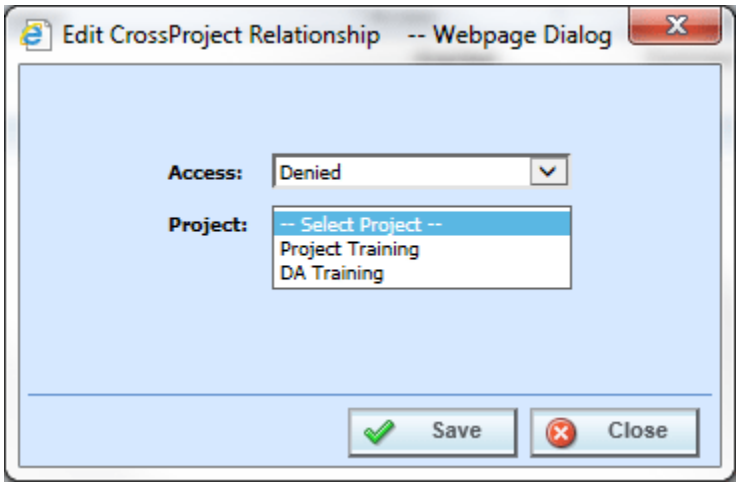


Figure 117 Granting Access to a Project

4. Select Granted from the Access option.
5. Select the project where you want to callout to. This is where the target program will be located.
6. Click Save to save your entry. The screen will close and the Manage Project Relationship screen will show that project access has been granted to the selected project.

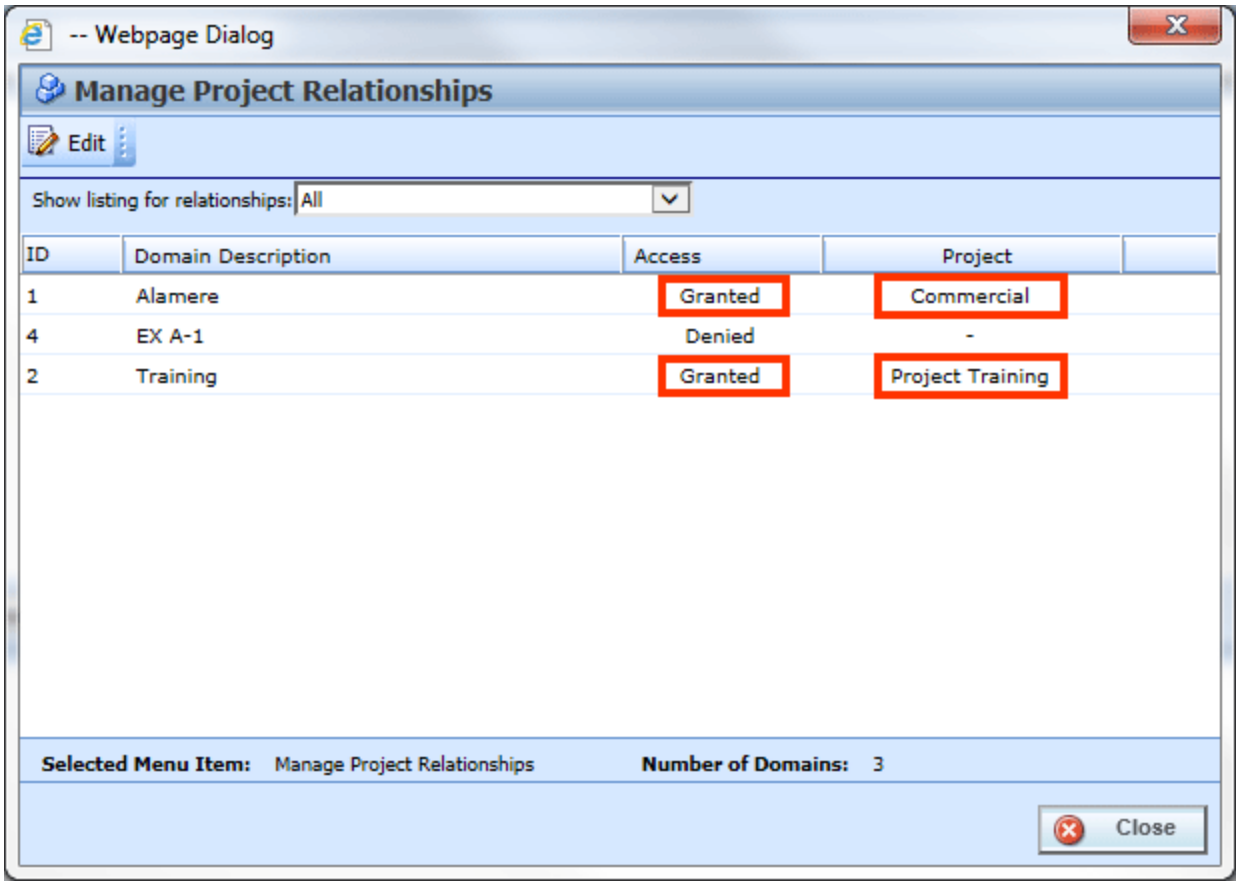


Figure 118 Completed Manage Project Relationship

After access has been granted, there are additional checkpoints that must be made before a callout can be implemented. For more information on working with Callouts, please see Callouts.

Available Domains

The available domains listed on the Manage Project Relationship screen are any domains with programs.

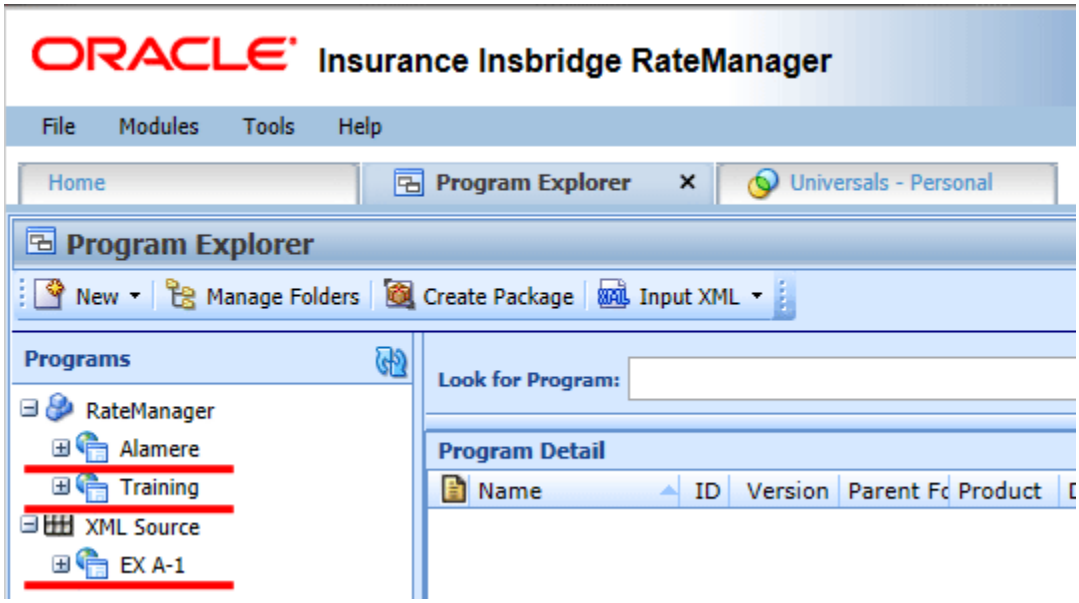


Figure 119 Available Domains for Project Relationships

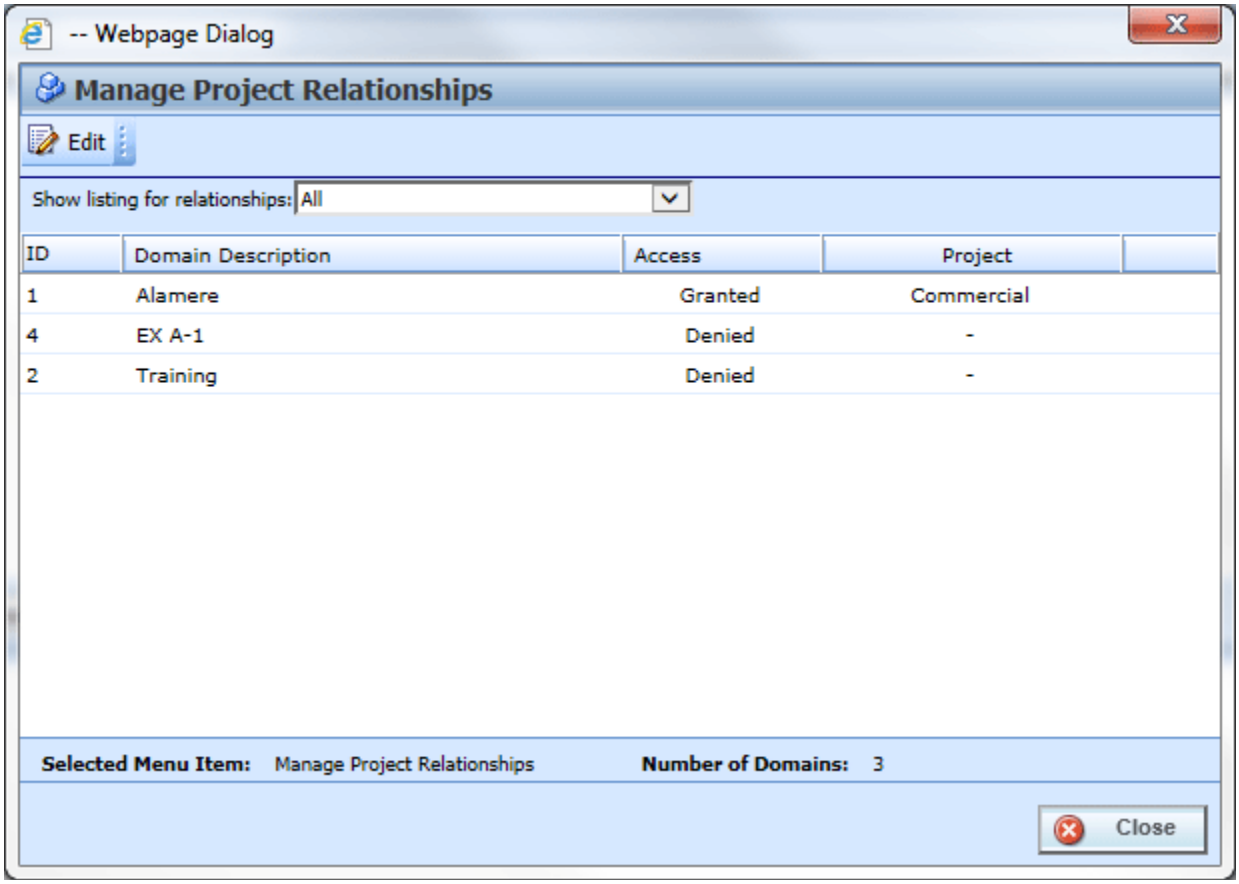


Figure 120 Domains for Project Relationships

The available projects that are listed on the Edit Manage Project Relationship screen are the projects with programs. If a project does not have any programs, it will not be listed on the Manage Project.

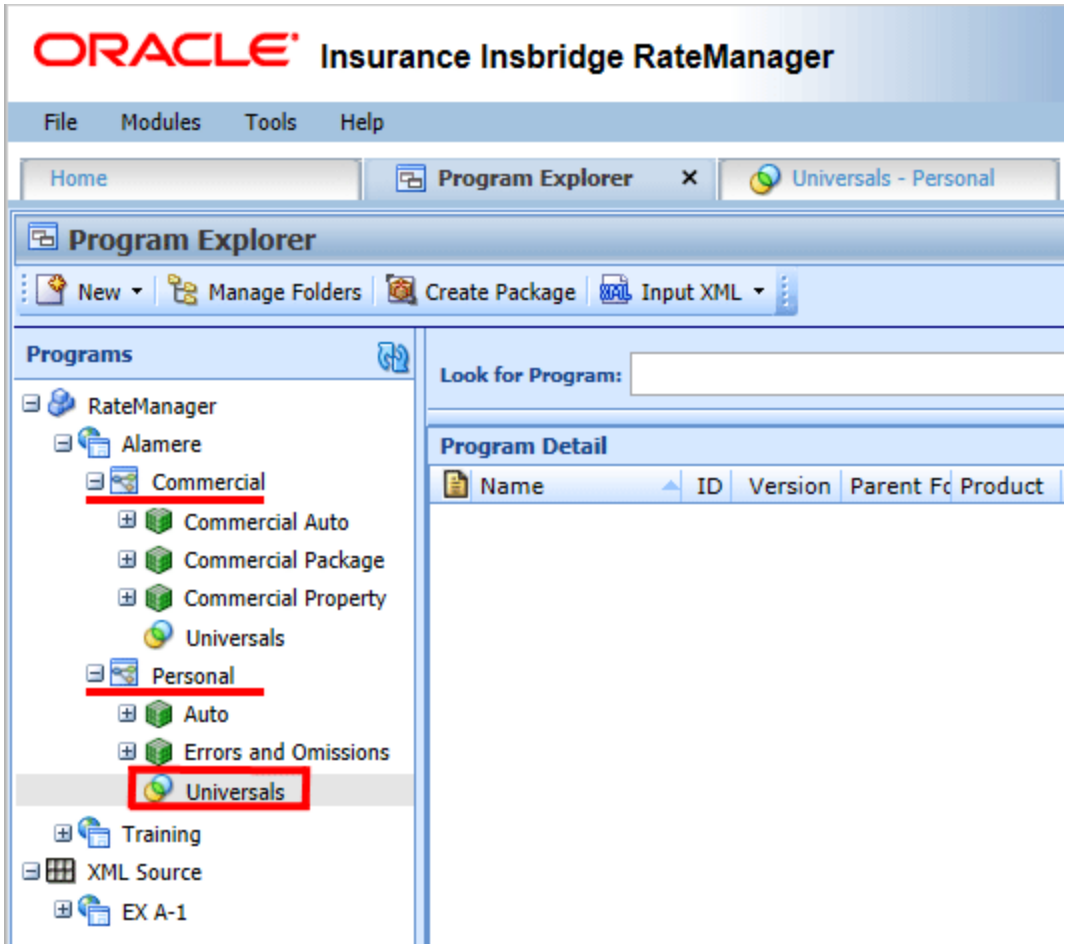


Figure 121 Available Projects for Selection

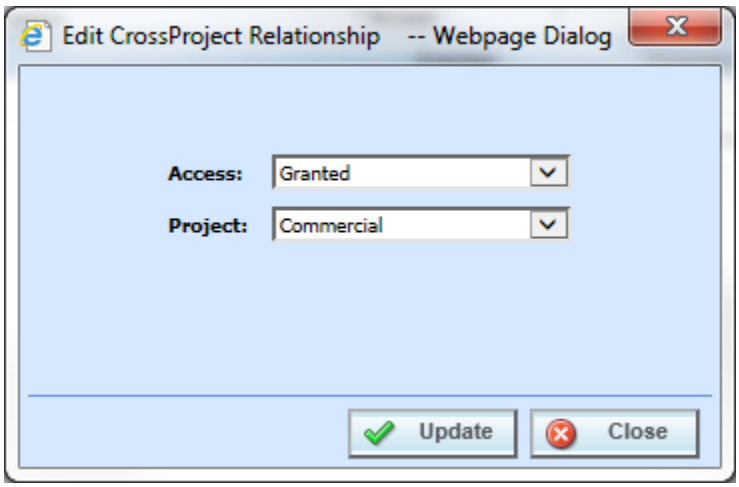


Figure 122 Selected Project

Since the Manage Project Relationship was accessed from the Personal Project, the Personal project is not listed.

Denying Access to Project Relationships

If access is no longer needed, you can deny access by changing the project access to Denied. To deny access, make sure you are in the project where you created the relationship.

1. On the Manage Project Relationships screen, select the listing you want to view; all projects, only activated projects or only deactivated projects. The screen will refresh with your selection.

2. Select the project you where you want to deny access. This is the location of the program you no longer want to callout to.

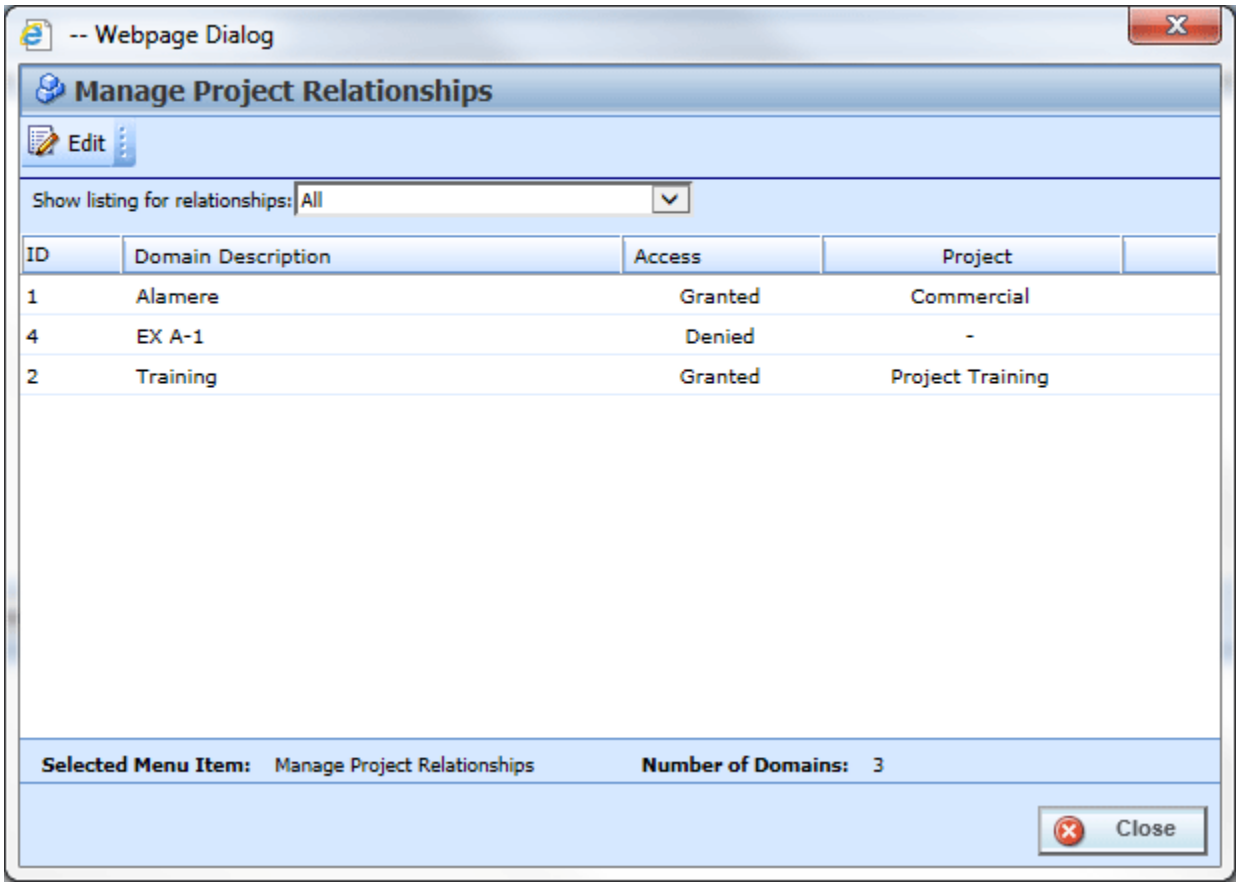


Figure 123 Editing Access for a Project Relationship

3. Click the Edit button.

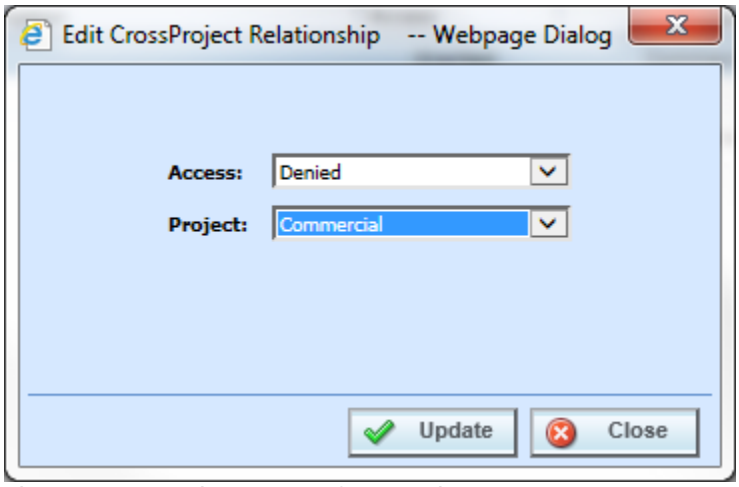


Figure 124 Denying Access for a Project

4. Select Denied from the Access option.

5. Click Save to save your entry. The screen closes and the Manage Project Relationship screen will show that project access has been denied to the selected project.

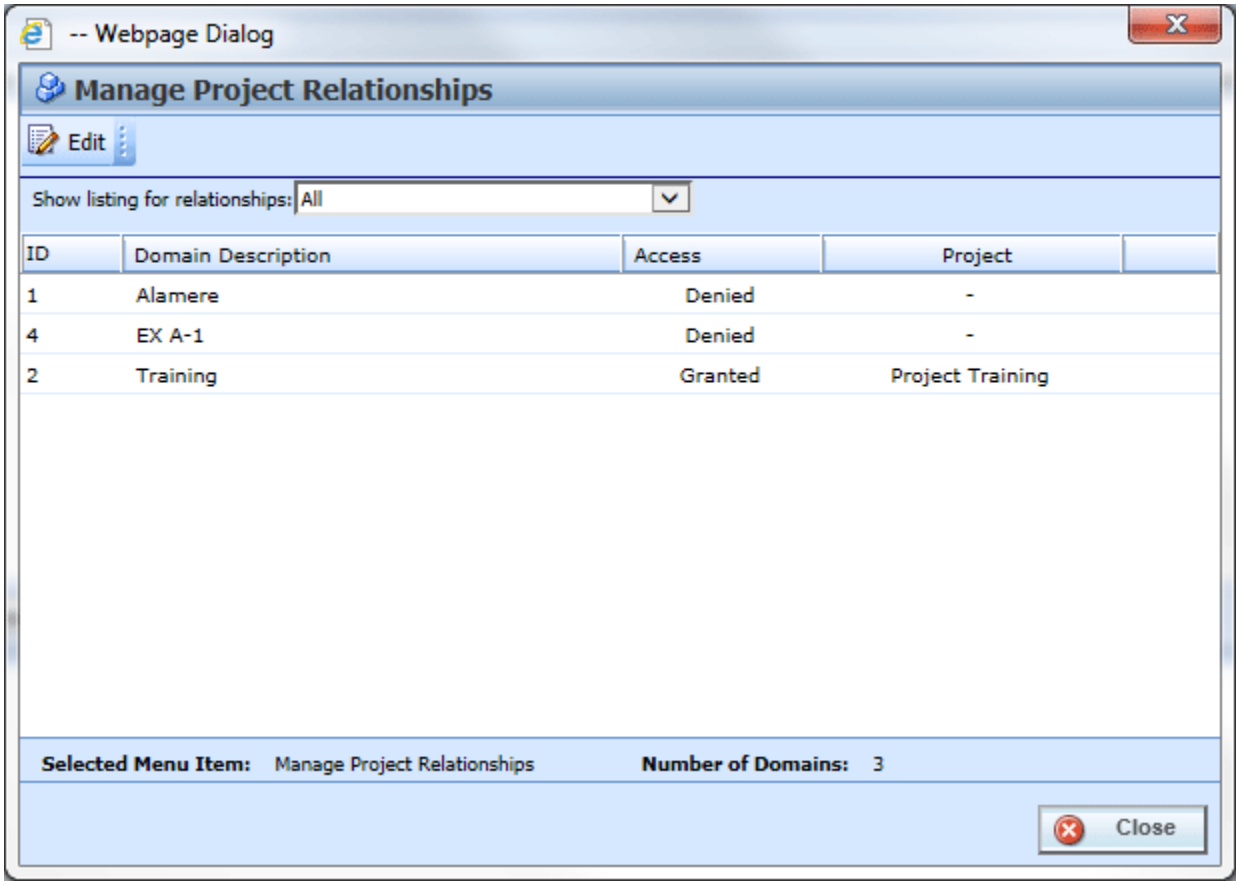


Figure 125 Updated Manage Project Relationship

Global Elements

Global variables, algorithms, driver assignments, constants, and filter rules can be created by the user in the RateManager domain type. In an Other domain type, these elements also can be imported or brought in from a template. Globals that are user created or imported are open for editing, copying, and deletion if they are not locked. Globals that were brought in from a template are locked at all times and cannot be unlocked. No editing or deleting is allowed. This is to ensure the integrity of the template. New revisions can be added to a template generated element. These new revisions are open for editing, copying, or deletion, unless the program version is locked.

Working with Global Variables

- At the global level, packages cannot be created and program testing cannot be done.
- Global and universal variables may be used in global table variables, global calculated variables, global algorithms, global driver assignments, or global filter rules. You cannot use any local elements. Fields, which are universal, can be used in global elements.
- Global elements are available for use by any of the programs under the product where the element is defined.
- Global elements have the active revision set at the program version level. The default active revision is revision 1. If there is another revision that you want to use, you must go into the program version and set the revision there. You cannot change the active revision in the global tab.
- Global variables can be copied to the local level. Calculated variables, table variables, algorithms, driver assignments, and filter rules can be copied from the global level into the local level of the same program or to a selected local program within the same product.

  Doing a global to local copy is a way of doing a “one off” variable that is only available to the program where it was copied to. For example, you have a global table variable with three criteria that is used in every program in the product. Then you create a smaller, more specific program, and you need to use this same table variable but you also need for it to have 4 criteria. Instead of copying it and leaving it in the globals area where it could be accidentally used, you can copy it to the local level of the smaller program. The table variable is open for editing, is not available to other programs in the product, and the global table variable does not have to be changed.
- Validations are created and managed at the universal level and are not available on the globals tab.

GLOBALS TAB

The Globals tab is available from the Program Explorer. This tab lists only the global elements associated with the selected product. The features and functionality on this tab also can be found on Program Details.

1. Select the domain type, domain, project, and product you need.

2. A Global icon is listed underneath the folders.

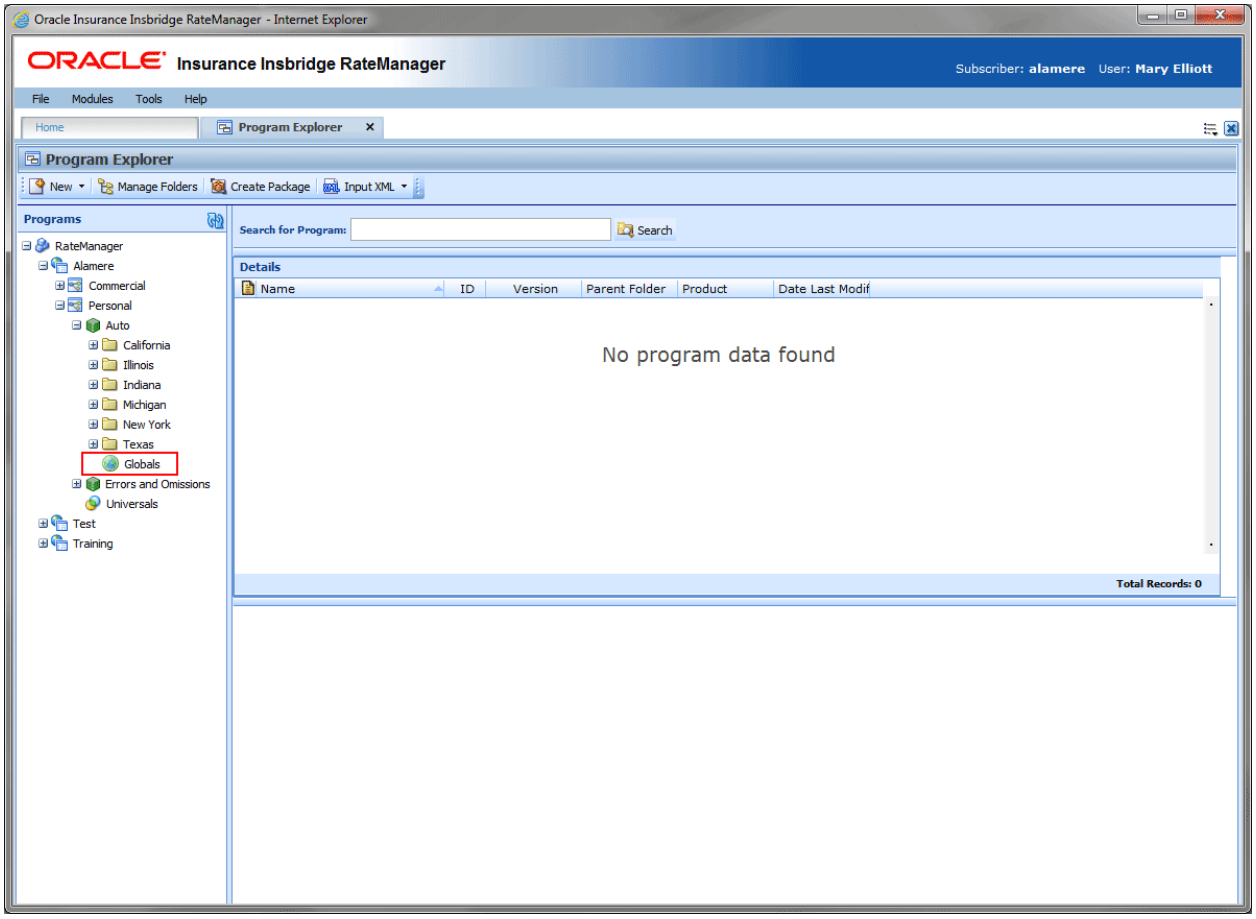


Figure 126 Selecting Globals for the

3. Double click the global icon and to open the Globals Tab.

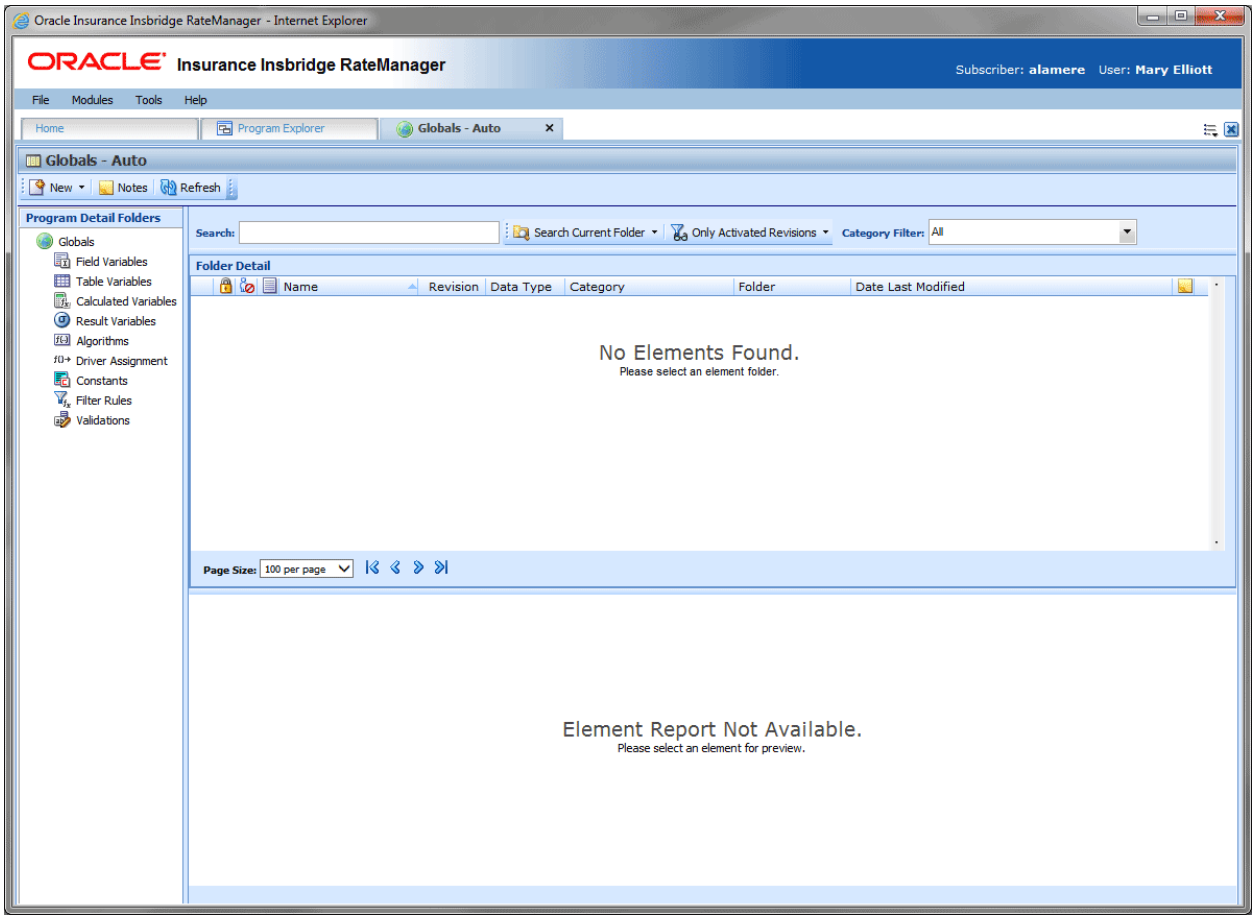


Figure 127 Global Tab

GLOBALS TAB FUNCTIONS

The Global tab functions like the Program Details tab.

New	**New:** Begins the process of creating a new global element. Result Variables, Calculated Variables, Constants, and Filter Rules allow you to click New from the element and open the corresponding tab or window. You must select the type of Table Variable, Data Source, Algorithm or Drivers Assignment you want to create from the menu.
Notes	**Notes:** Opens the Notes screen for the selected element.
Refresh	**Refresh:** Refreshes the screen with the current information.

Globals Tab Screen Details

The Global tab functions like the Program Details tab with the exception of setting revisions. Revisions cannot be set at the global level or universal level.

Chapter 12

CATEGORIES

The category concept is central to program loading in RateManager. Categories define fields, variables, algorithms and the overall structure of a program.

A category is a group of information that is specific to an item. For example, an auto policy would have categories of Policy, Driver, and Vehicle. A home policy would have a Policy and Dwelling category.

When categories are defined in RateManager, their relationship to existing categories is also defined. For example, in an auto program we may define Accidents and Convictions as belonging to Driver. Driver is the parent category of the Accidents and Convictions categories.

When variables and algorithms are created within RateManager, they are always assigned a category. The purpose of the category is to define the number of iterations the variable or algorithm should run. In the auto example, a Driver level category variable would run once for every driver on the policy. In the home example, a Dwelling level variable would run once for every dwelling on the policy.

When to Use a Category

Use a category to group a set of fields together, so that the set runs for every instance of the group. For example, in calculating Driver Accident Points with a field of ViolationCode, you have to run a variable for every instance of ViolationCode for every driver. In this case, you could make a category of Violations whose parent category is Driver. In the Violations category, you would create a field of ViolationCode. When you create a variable like Accident Type, which is based on ViolationCode, the rating engine would look up an Accident Type for each ViolationCode passed into the system.

The following excerpt from an input file would represent a single driver with two accidents:

```
<ci="1" desc="Driver">
   <mi="23" d="BirthDate" v="09/13/1985"/>
   <mi="25" d="Gender" v="M"/>
   <ci="6" desc="Violations">
      <mi="102" d="ViolationCode" v="12"/>
   </c>
   <ci="6" desc="Violations">
     <mi="102" desc="ViolationCode" v="34"/>
   </c>
</c>
```

Navigating to Categories

Categories are project specific and are not an option on the Program Explorer tab or the Program Details tab. You must select the project where you want to view or manage categories and then click the universals icon. This opens the universals tab where you can access the Categories screen.

1. Modules→Program Explorer. This opens the Program Explorer tab.

2. Select the domain type→domain→project→product→universals to open the universals tab.

3. **Categories** are available from the menu bar.

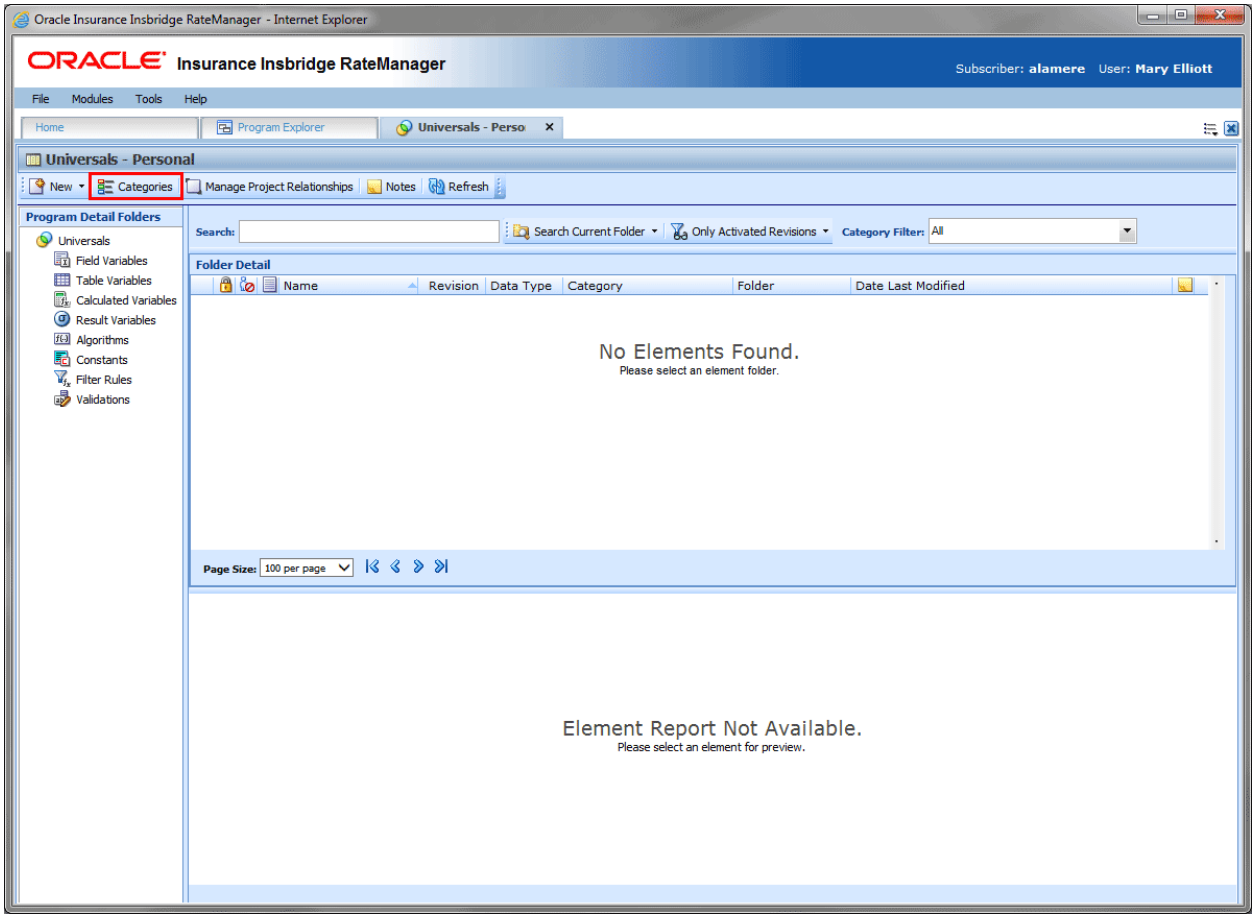


Figure 128 Categories Option on the Universals Tab

WORKING WITH CATEGORIES

The Categories screen lists all the categories for the project. Every project has a system created Policy category (XML ID 0) that cannot be edited or deleted. New categories can be created under policy or at the same level.

If the driver assignment module has been enabled, the auto product categories are automatically added. The categories Driver, Vehicle, and Driver-Vehicle cannot be edited, deleted, or moved and can only be used in the auto category. The Driver-Vehicle category cannot have any other category underneath it.

The XML IDs for these categories cannot be changed and are always the same regardless of project, product, or program version. The XML ID for:

- Driver category is 1
- Vehicle category is 2
- Driver-Vehicle the category is 3

If the auto product has not been selected then the driver, vehicle and driver/vehicle categories are not listed.

You can create as many categories as you need. Categories utilize a parent/child relationship that allows for categories to be placed underneath each other or on the same level. This creates a multilevel set up where you can place an unlimited number of categories underneath the top level or any other category.

Categories that are locked cannot be edited or deleted. Locked categories have a lock icon.

NOTE: *In the auto product, your initial working category options are Policy, Driver, Vehicle and Driver-Vehicle. To create a variable that used both driver and vehicle fields, you would need to define this variable in a Driver-Vehicle working category.*

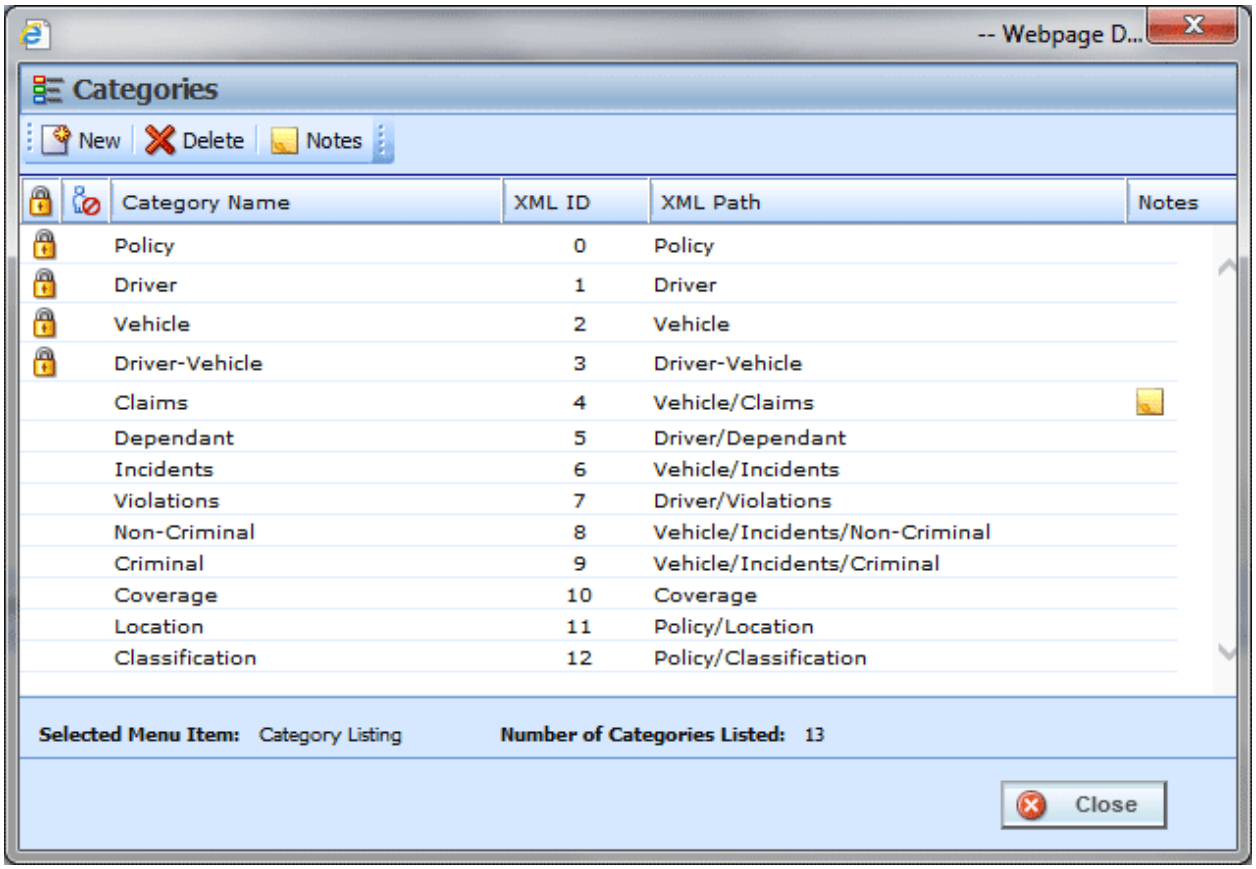


Figure 129 Categories Screen

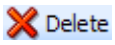
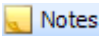
NOTE: *The Policy category (XML ID 0) always exists. It cannot be edited or deleted.*

The Categories screen shows the XML ID and parent category. From this screen you can:

- Create a Category
- Edit a Category
- Delete a Category
- View History: See View History.
- Create, edit, and delete Notes: See Notes.

CATEGORIES FUNCTIONALITY

The **Menu Bar** is located at the top of the screen. The menu bar contains the functions you can perform on this screen.

	New: Creates a new category.
	**Delete:** Deletes the currently selected category.
	**Notes:** Opens the Notes screen for the category.

CATEGORIES DETAILS

Categories can be sorted by the Category Name and XML ID. To change the order, click the column you want to sort by. The default view is numerically (lowest to highest) by XML ID. To sort by a different column header, click the column header you want and the current results will be resorted. You can sort results as many times as you want. Sorting does not filter results. It only rearranges the order in which they are displayed.

Category Listing

The category listing screen shows a listing of categories for the project. To open a category, double-click it.

Icon: There are two icons:

Locked. A lock in this part of the screen indicates that the category is locked. Category locks from templates may have a different lock icon.

In-Use. Indicates that the category is being worked on by another user.

Category Name: Lists the name of the category.

XML ID: This number is automatically assigned by RateManager at the time of creation unless a number has been entered in the XML ID field. This number is used to distinguish between categories when rating. XML ID numbers cannot be edited.

XML Path: Shows the parent category of the category.

Notes: A note icon indicates a note for this category.

Right Click Options

The right click menus depend upon the category. Policy, driver and vehicle have three options, and driver-vehicle has two options. All User created categories have five options.

To view the right click menu, select the category and right click.

- **New:** Opens the New Category screen where you can create a new category. New categories cannot be created under Driver-Vehicle.
- **Edit:** Opens the Edit Category where you can edit the selected category. Policy and Driver-Vehicle cannot be edited.
- **Delete:** Deletes the currently selected category. Categories that are in use cannot be deleted. Policy, Driver, Vehicle, and Driver-Vehicle cannot be deleted.
- **View History:** Opens the Audit History of the category.
- **Notes:** Opens the Notes screen where you can view and manage notes for the selected category.

When a program is locked, the categories used in that program are locked also. Locked categories cannot be edited or deleted.

Creating a Category

When creating a category, there is a top-level parent category. All categories are created under Top. If you have not created any categories, you can add a new category under Top which would be placed on the same level as Policy or create a new category under Policy. If there are multiple categories already in the system, you can select the category pathway where you want to create a new category.

Naming standards can be applied to categories. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the category and for case enforcement to be applied to any newly created category or any existing category that is being edited. If a naming standard is in place, the pattern is displayed in the category name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented.

NOTE: *Oracle Insurance recommends that the Category Name not contain any spaces and that the Category Names be unique, regardless of the pathway.*

NOTE: *Once you have created a new category, it will be available from the working category drop down boxes in the variable and algorithm screens.*

1. Select the Project where you want to create a new category.
2. Click the universal icon to open the universal tab for the project.
3. Open the **Category** screen.
4. Click **New**. You also can highlight most existing categories and right click. Select **New** from the menu.
5. A **New Category** window is shown.

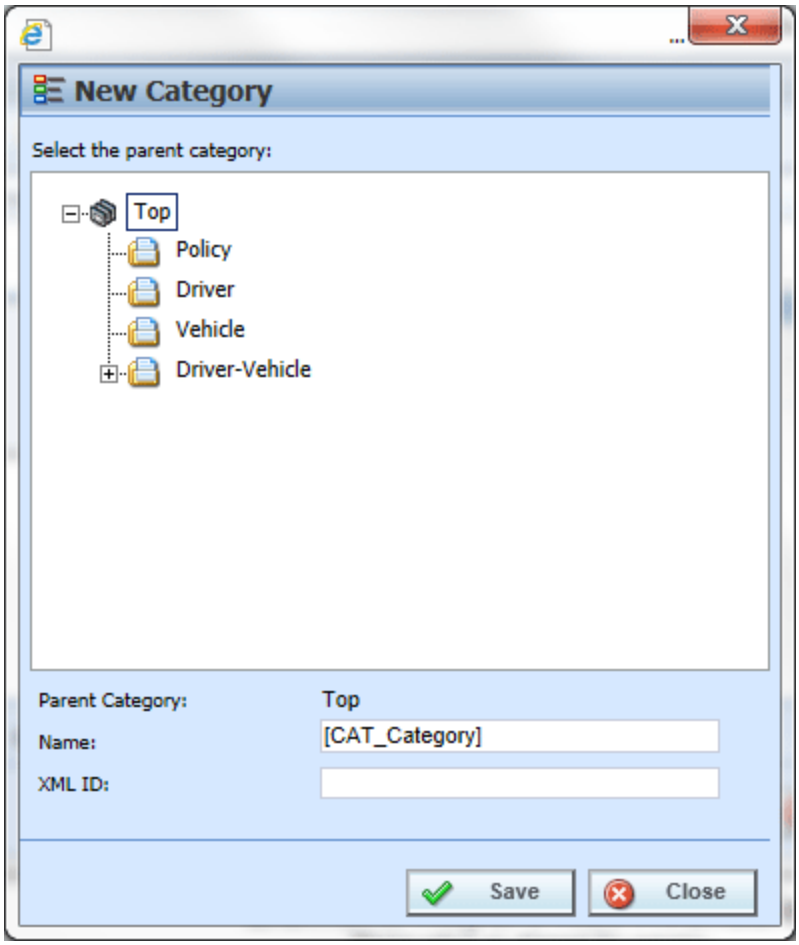


Figure 130 Creating New Category

6. Select the pathway for the new category. The Parent Category field shows the currently selected path. On the tree view, the final destination for the new category has a box around it.

7. **OPTIONAL:** If you have to create a category with a specific XML ID, you can enter it. If the number is already in use, you will receive an error message.

8. Enter in a **Name**. Click **Save** to save your category. The screen will refresh and the new category will be displayed in the Categories screen.

Editing a Category

Editing a category changes the category's name or changes the parent category.

Changing a parent category changes the category location within the same project. When you move a category, all associated subcategories are moved as well. There is no limit to the number of times a category can be moved. The Policy, Driver, Vehicle and Driver-Vehicle categories are always a top level category and cannot be moved underneath another category. Moving is performed on the Edit Category window.

Notes on Editing a Category

- You cannot change the XML ID of a category. If you do not enter a value, RateManager automatically assigns the XML ID when the category is created. Once assigned, it cannot be updated.
- System categories that are grayed out cannot be moved. Name changes may be allowed.
- Categories that are locked are not editable.

WARNING: ***If you have already started creating your program, changing the parent category is not recommended.***

This is especially true for templates. Changing a category on an updated template may cause it to be rejected by the user who is trying to apply it.

1. Navigate to the **Category Screen** for the project that contains the category you want to edit.
2. Select the category you want to edit or move and either double-click it or right click it and select **Edit** from the menu.
3. The **Edit Category** window is displayed.

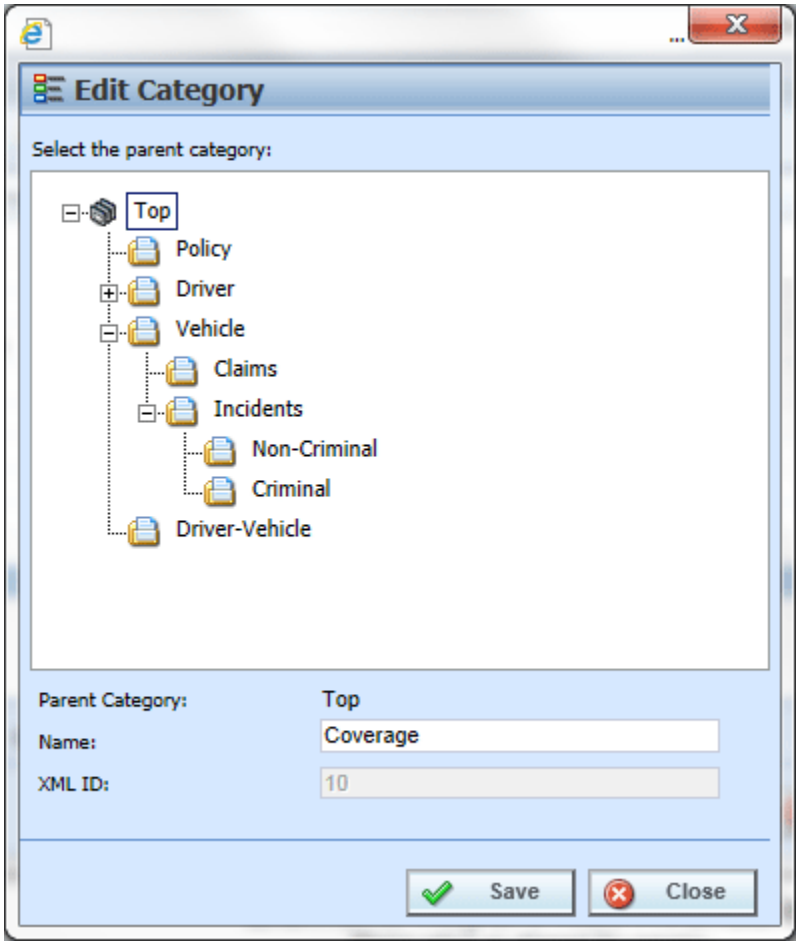


Figure 131 Editing a Category

4. To change file location, choose a new parent category. The folder you have selected to move is displayed in the Name field but will not be displayed in the tree structure. Double click the folder where you want to move to. The Parent Category field will reflect the new path for the category.

5. Make any needed **Name** changes.

6. Click **Save** when you are finished to refresh the category listing.

Click **Close** to go back to the Category screen without saving changes.

Deleting a Category

If you no longer require a category, you can delete it. When you delete a category, the reference to the XML ID also is deleted and is available to be assigned to a new category.

In the Auto product, the categories Driver, Vehicle and Driver-Vehicle cannot be deleted.

Notes on Deleting a Category

- Categories being used by an element cannot be deleted.
- System categories that are grayed out or have a lock icon cannot be deleted.

To Delete a Category

1. Navigate to the **Category** screen for the project that contains the category you want to delete.
2. Select the category you want to delete and either click **Delete** or right click and select **Delete** from the menu.

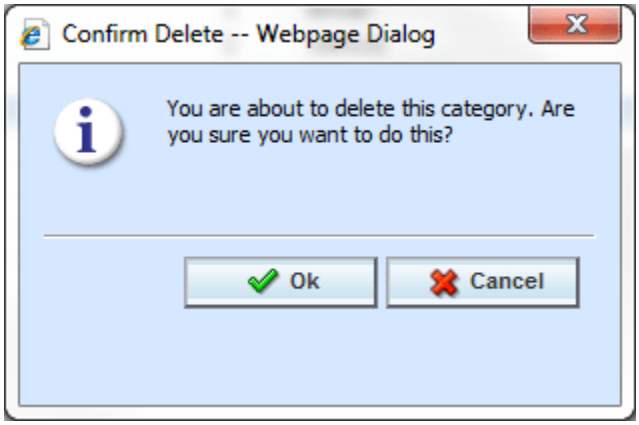


Figure 132 Deleting a Category

3. The category is deleted and the category listing will be refreshed. If the category is still being used by an element, you will receive an error message.

FIELDS

Fields pass information into the SoftRater engine. Fields are product specific and managed at the universal level. There are two areas to only view fields, either from the Globals tab or from the Program Details tab. When selected, fields are listed in the Folder Details area.

There are user created fields, fields brought in from a template, and system fields. User created fields are open and can be edited, copied, or deleted unless locked. A user created field that is locked only allows for copying. Fields that were brought in from a template are locked at all times and cannot be unlocked. No editing or deleting is allowed. This is to ensure the integrity of the template. Copies can be made of template generated fields.

Fields can have a list of values (validations) assigned. Lists of values are user created tables that hold values and labels that describe the values. Lists of values are assigned to field variables then used in other variables. The purpose of a list of values is to provide the ability to select values based on their label/description.

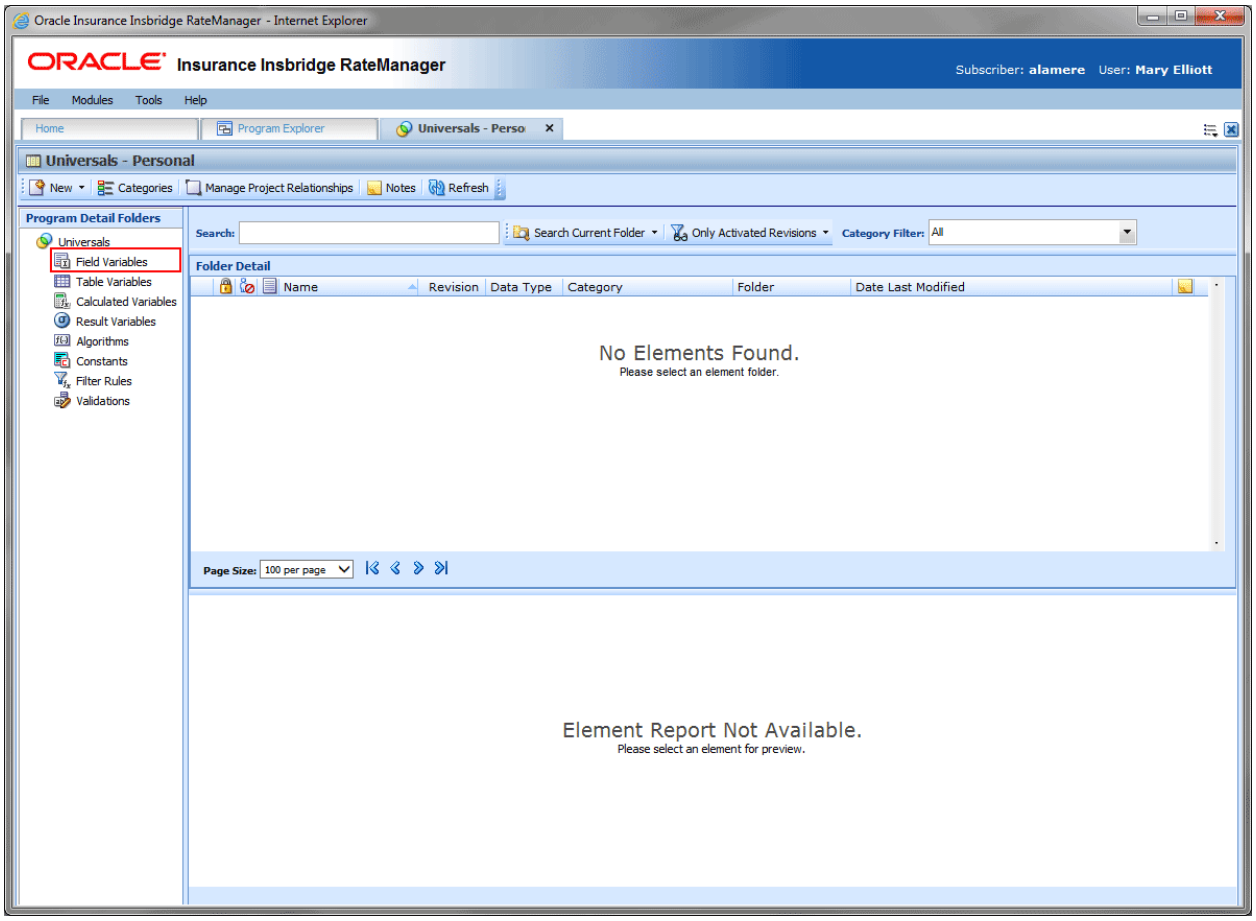


Figure 133 Selecting Fields

System generated fields are grayed out and cannot be edited, copied or deleted. The auto product has 39 system created fields that are needed for driver assignment. For example, VehPrincipallyDriven (XML ID=299) is a system field. These fields cannot be edited, copied, deprecated, or deleted. It is recommended that you do not create another field using a system field name.

Navigating to Fields

Fields are not an option on the Program Explorer tab.

1. Modules→Program Explorer to open the Program Explorer tab.
2. Select the domain type→domain→project→universals where you want to manage a field.
3. On the universals tab, select Fields from the Program Detail Folders column.

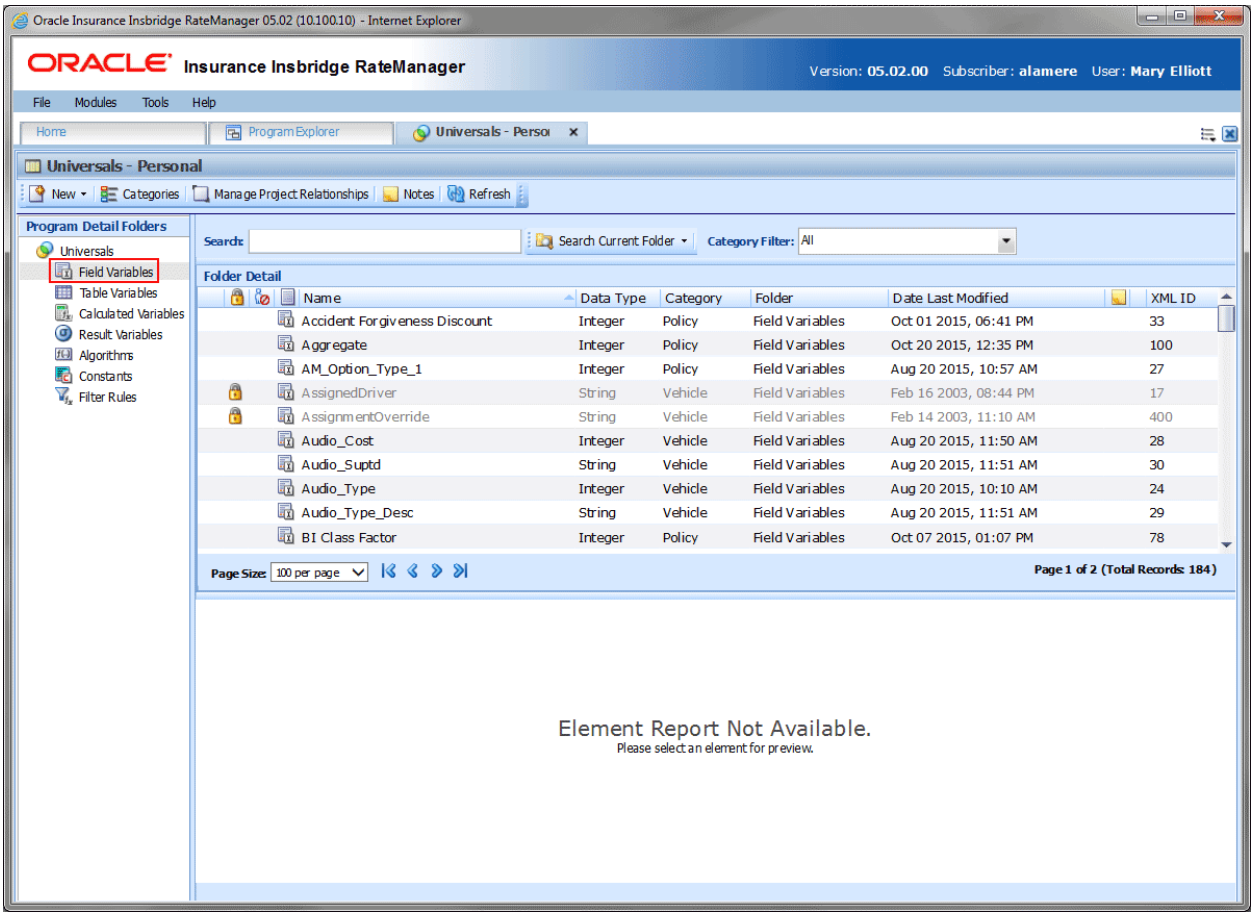


Figure 134 Field Listing

FIELDS FUNCTIONALITY

Functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any field. The functions you can perform depend on the status of the field. Locked fields have fewer options. There are icons that can tell you at a glance what the status of the field is and if a validation has been assigned.

Icon	What it Indicates

	Fields. Fields are project specific and can be created at the universal level, or from another program element. All fields are displayed on the Program Details→Fields tab. The same field listing also is available for Project→Universals tab or Product→Globals tab. The options new, edit, copy and delete are available only on universal tab. Deprecate is available from either the universal or global tab.
	Deprecated Field. Fields that have been deprecated have a field icon with a red arrow. Deprecated fields can be restored if necessary. Fields can be deprecated from the universals tab or the globals tab.
	Locked Field. Locked fields are not open for editing. A lock icon indicates that this field is locked by either a program or is system generated. System generated fields are not open for copying, editing, or deleting. Program locked fields can be copied. If you have to edit a program locked field, you must return to the program and unlock it. This unlocks the elements and changes can be made. If the field is locked because it is system generated, you cannot unlock the field.
	Template Lock. If the field is locked because it came from a template, you cannot unlock the field. You can copy the field or create a new field for use in the program if necessary.
	In-Use. Fields that are in use are currently being worked on by another user in the system are not open for editing, copying, or deleting. The field will be available when the user exits the field.

Fields with validations have a slightly different icon. The data type of the validation changes the icon to indicate that the field has a validation assigned to it.

Icon	What it Indicates
	Date Validation. The validation is a date data type.
	Decimal Validation: The validation is a decimal data type.
	Integer Validation. The validation is a integer data type.
	String Validation. The validation is a string data type.

NOTE: *You can search Fields by name or by XML ID number. Fields being searched by XML ID must include the text "ID:" The format is: "ID:(number)".*

FIELD DETAILS

The screen details are the same for the Universal tab, Global tab and program details. For more on screen detail descriptions please see Program Details→Screen Details.

Right Click Menu Options

To view right click menu options, highlight the field you want to view and right click. Unlocked field variables have seven right click menu options:

- **Edit:** Edits the field.
- **Copy:** Brings up the copy screen for the field.
- **Delete:** Removes the field from the list. A warning message is displayed to confirm the deletion. You must have permissions to delete.
- **Deprecate:** Removes the field from use by other elements. Not present on deprecated fields.
- **Restore:** Deprecated fields can be restored with this option. Only present on deprecated fields.
- **View Where Used:** Displays an information screen that details where the variable is being used. See View Where Used for more information.
- **View History:** Displays the Audit History report. See View History for more information.
- **Notes:** Enter, edit, and view notes for the field.

Template locked field variables and program locked fields have four options, deprecate, View Where Used, View History, and Notes.

CREATING A FIELD

In order to create a new field, you must have your categories set up. RateManager automatically has a Policy category for each project. Fields can be system defined, template generated or created by you.

Template generated fields are locked and not open for editing, or deletion. A user created field can be added to a template generated program if needed. The new variables are open for editing, copying, deprecating, or deletion.

Fields should be unique at the category level. This means that you should use unique field names in every category. For example, if you need a BaseRate field for Policy and Driver, you can create a BaseRate field for the policy category and a BaseRateDR field for the Driver category.

Naming standards can be applied to system elements. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the element and for case enforcement to be applied to any newly created element or any existing elements that are being edited. If a naming standard is in place, the pattern is displayed in the element name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented for all elements.

1. On the universals tab, select Fields from the Program Detail Folders column. Click New in the top bar menu. This opens the New Field Variable window.

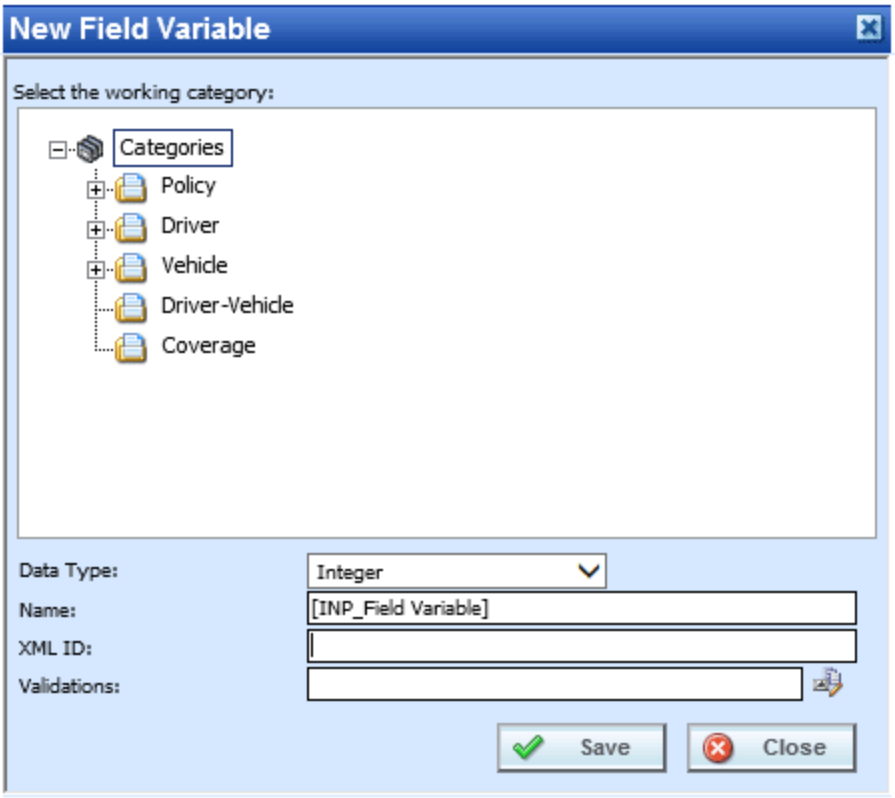


Figure 135 New Field Variable Window

2. Double click the category where you want the new field to be placed. If Policy is the only category, you may need to return to categories to create the one you need. On the tree view, the final destination where the new field will be placed has a box around it.

3. Select a Data Type from the menu.

4. Enter in a Name. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

5. **OPTIONAL:** If you have to create a field with a specific XML ID, you can enter it. If the number is already in use, you will receive an error message. If you do not enter a number, RateManager automatically assigns one.

6. **OPTIONAL:** If a list of validation is needed, click the validations icon at the end of the row. Select the validation you need. For more on validations, please see Chapter 1 Validations. If a validation is not needed, leave the row blank.

7. Click Save to save your field. The screen refreshes and the new field is displayed on the universals→field tab. An XML ID is automatically assigned for any new field variable if a number is not entered.

Custom XML ID

Custom XML IDs are an optional feature where you can enter in the exact XML ID you need. A custom XML ID can be any positive number 1 – 5 digits long. Zero (0) is not allowed as the first digit. Custom XML IDs must be unique to the project but do not have to be sequential. You can enter an ID of 97 for a field and then enter 530 for the next field. You do not have to create the next field as ID 98. If an ID is being used, you will receive an error message.

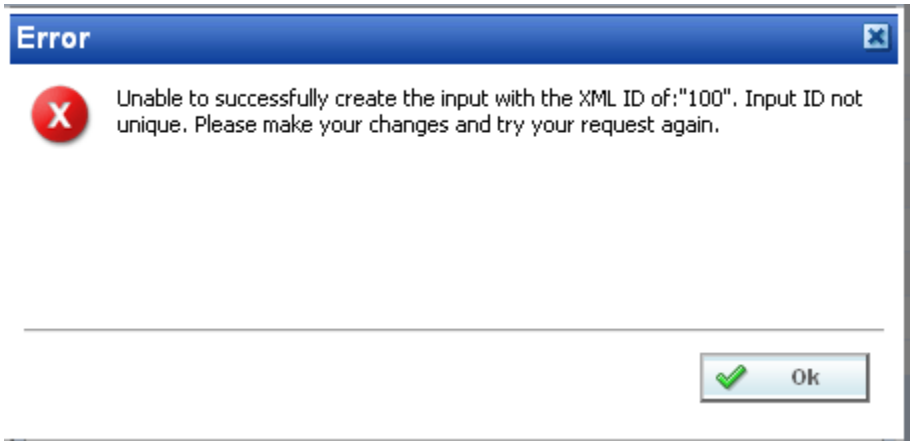


Figure 136 Custom XML ID Error

Custom XML IDs allow you to match fields across projects. For example, if you have a field that every project is going to use, such as Effective Date, you can create the same field using the same ID in every project. This can be useful when creating programs for cross project P2P.

When entering a custom XML ID, please verify your work prior to saving. XML IDs cannot be edited. If you have made a mistake, you will have to recreate the category or field. The incorrect entry can be deleted if necessary. Deleting a category or field releases the XML ID to be used again.

Custom XML IDs cannot be created on copied fields. Copied fields have system generated XML IDs.

Custom XML IDs are available in Categories and Fields.

Adding a Field from another Area

A new field can be created while entering or editing a table or calculated variable, algorithm, or driver assignment without leaving the Edit tab. The category you can add the field to is determined by the Available Variables, Functions and Constants selection. Except for the XML ID, the new field can be edited on the universal→field tab, if needed.

All new fields created from another area can be found in the universals tab.

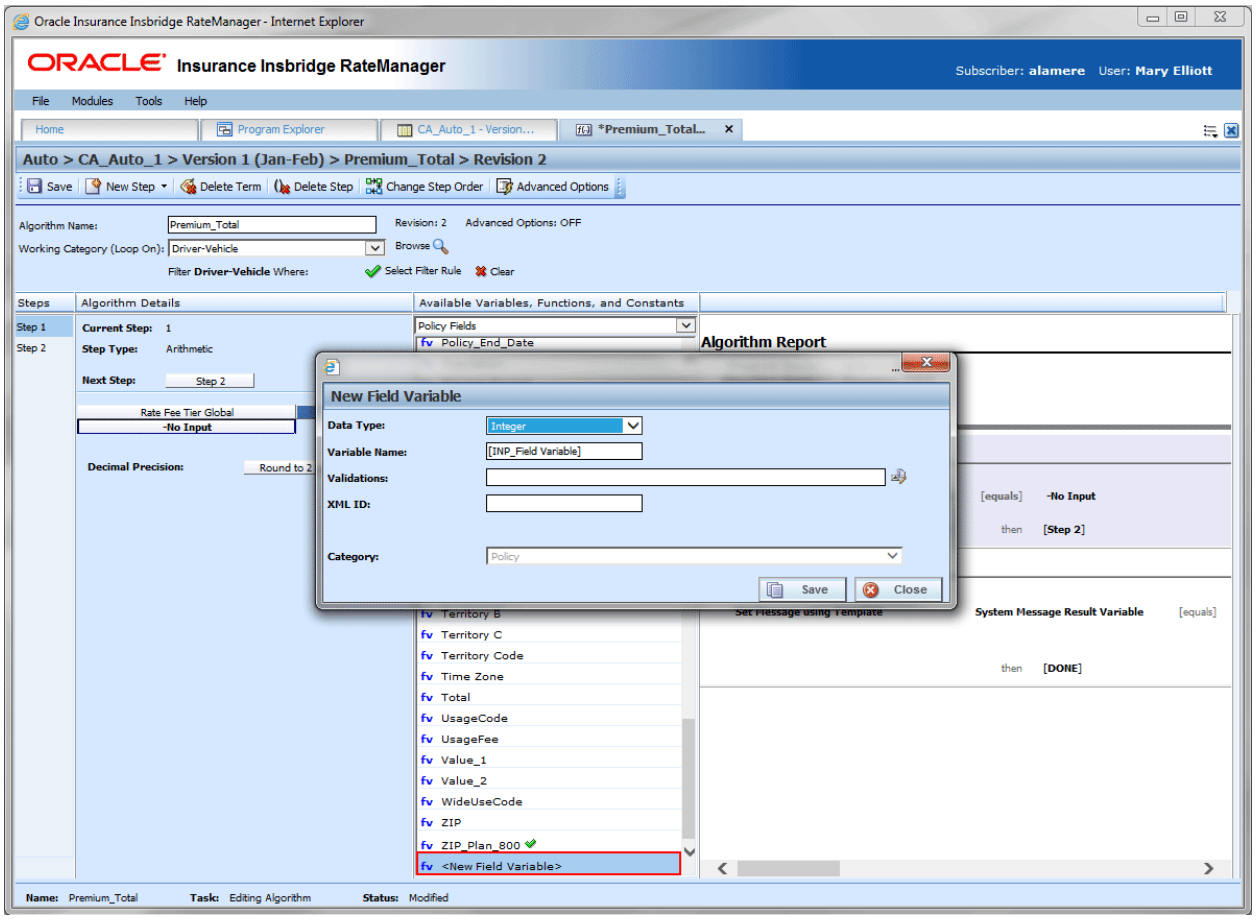


Figure 137 Adding a Field Variable

1. Select the field category you need in the information section of the variable.
2. In the Available Variables, Functions and Constants listing scroll down to the bottom of the fields list in the category you need. Select **<New Field Variable>**. The New Field Variable screen is displayed.
3. Enter in a Variable Name. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.
4. **OPTIONAL:** If you have to create a field with a specific XML ID, you can enter it. If the number is already in use, you will receive an error message. If you do not enter a number, RateManager automatically assigns one.
5. **OPTIONAL:** If a list of values validation is needed, click the validations icon at the end of the row. Select the validation you need. For more on validations, please see Chapter 1 Validations. If a validation is not needed, leave the row blank.
6. Select the Data Type.
7. Click **Save** to close the New Field screen. Your new field variable is available for selection.

Editing a Field

Unlocked fields can be edited at any time. If the field is locked because the program is locked, you must return to the program and unlock it. Once unlocked, changes can be made. If the field is locked because it came from a template, you cannot unlock it. System generated fields are grayed out and are not available for editing at any time. XML IDs cannot be edited.

NOTE: *If a change is made to a field, the change affects all programs that use that field.*

1. Either double click the field you want to edit or right click and select Edit. This opens the Edit Field Variable window.

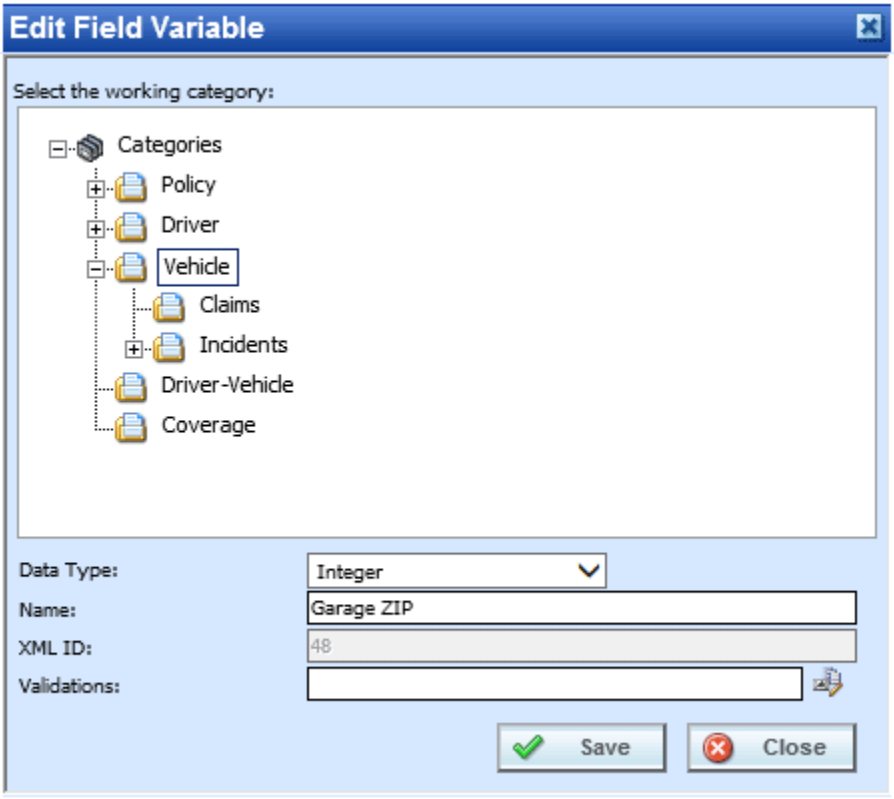


Figure 138 Editing Field Variable

2. To change the Category, double click the category where you want the field to be located. On the tree view, the final destination where the field will be placed will have a box around it.

3. Make any changes to the Data Type and Name. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

4. Click Save to save your changes.

Copying a Field

Fields can be copied at any time. The field is copied with the exact same data type, category, and validation. To make a change to the field, you must go into the edit screen. System generated fields cannot be copied. Names cannot be duplicated within the same category.

Copied fields receive an auto generated XML ID from the system. You cannot enter an XML ID for a copied field. If you have to enter an XML ID, you must create a new field.

1. Right click the field you want to copy. Select Copy. The Copy Field Variable window is displayed.

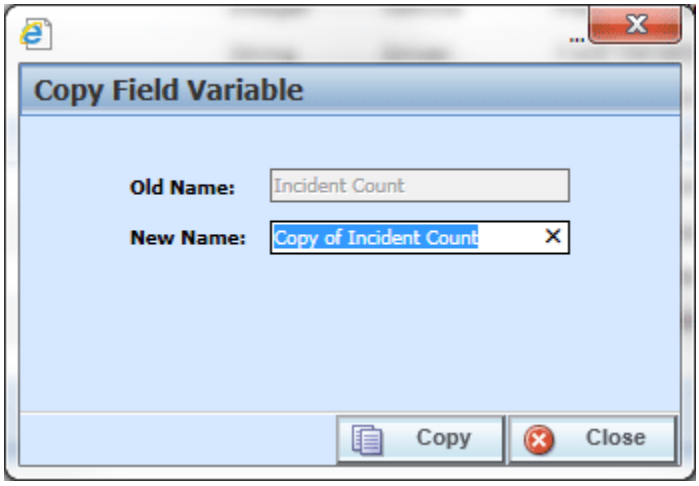


Figure 139 Copying Universal Field Variable

2. Enter in a New Name. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Click Copy to copy your field. The screen refreshes and the copied field is shown on the universal screen→fields listing. An XML ID is automatically assigned for the field.

Deprecate a Field

Any unlocked field variable can be deprecated from the universal tab or the global tab. Deprecation allows you to hide a field from being used in any new or edited variables without removing it from variables where it is currently being used. You can use this feature when you want to replace an existing field without having to replace it in multiple places. System generated fields cannot be deprecated.

1. Fields can be deprecated by right clicking the field and selecting Deprecate.

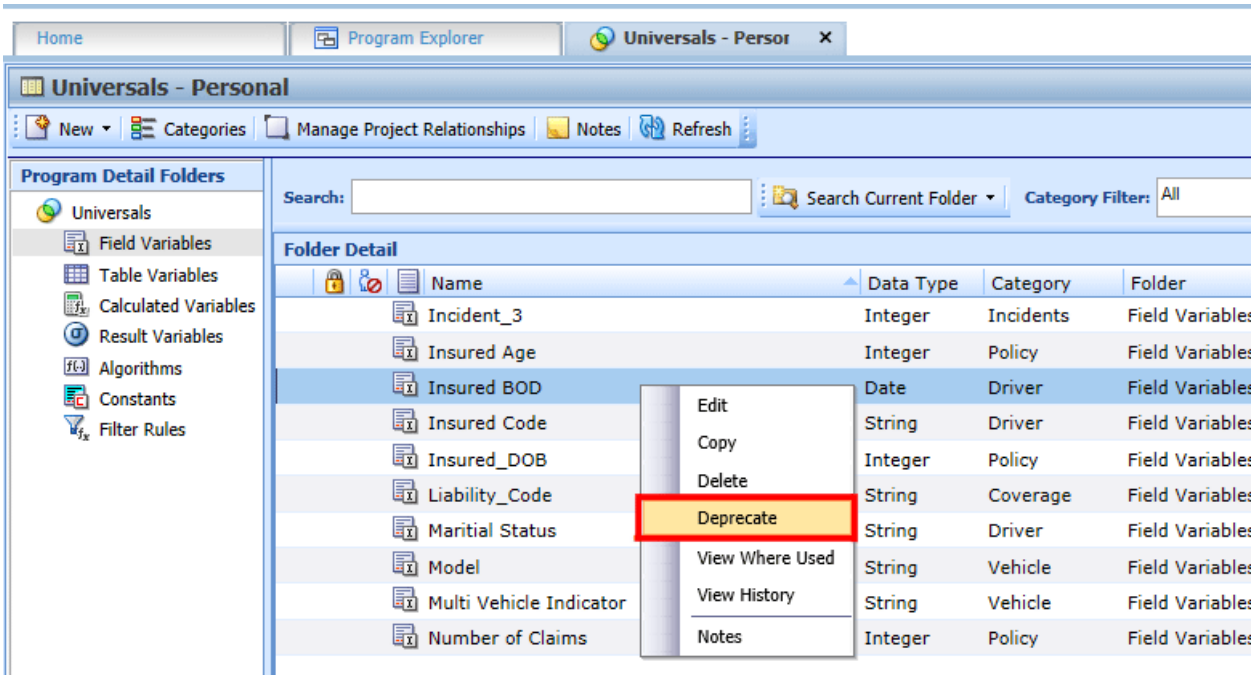


Figure 140 Deleting a Field

2. A confirmation message is displayed.

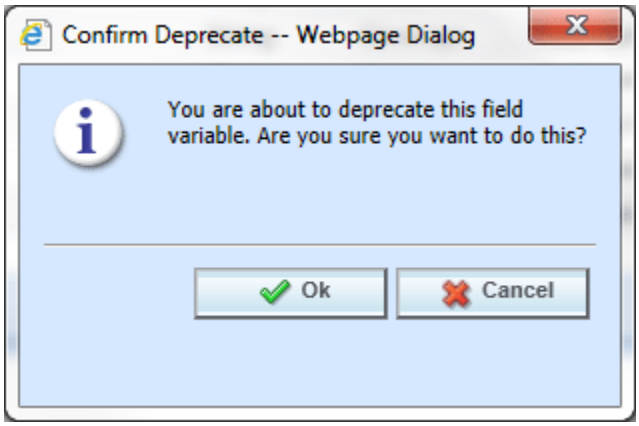


Figure 141 Deprecate Field Confirmation

3. Click **OK** to deprecate the field or **Cancel** to cancel.

Deprecated fields are not displayed in the available variables listing on table or calculated variables, algorithms, driver assignments, or filter rules. Deprecated fields can still be edited, copied or deleted.

Restoring a Field

Any deprecated field variable can be restored from the universal tab or the global tab. Restoring a field returns it to full usage. System generated fields cannot be restored.

1. Fields can be restored by right clicking the field and selecting Restore.
2. A confirmation message is displayed.

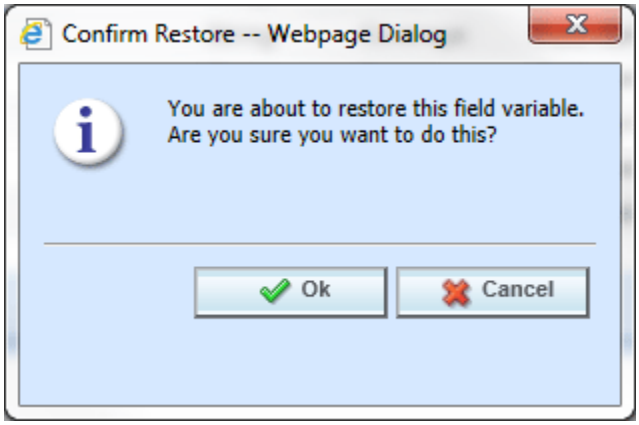


Figure 142 Restore Field Confirmation

3. Click **OK** to restore the field or **Cancel** to cancel.

DELETING A FIELD

Any unlocked field variable that is not in use can be deleted. Fields that are in use, locked, or system generated cannot be deleted. If you try to delete a field that is unavailable for deletion, an error message is displayed informing you where the fields are being used. System generated fields are grayed out.

NOTE: *Take care when deleting Fields, this action cannot be undone*

1. Fields can be deleted by right clicking the field and selecting **Delete**.

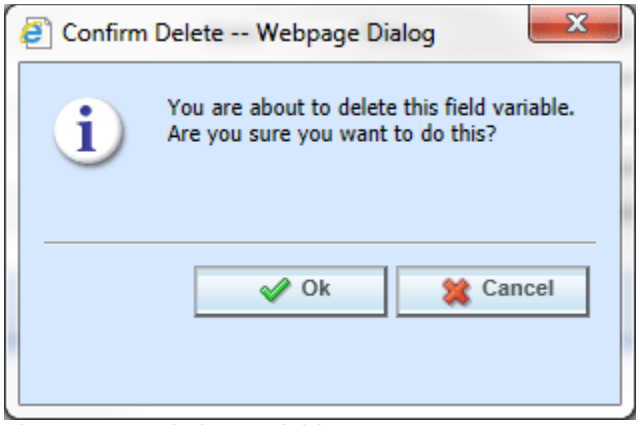


Figure 143 Deleting a Field

2. A confirmation message is displayed.

3. Click **OK** to delete or **Cancel** to cancel. If the field is being used, you cannot delete it. A message is displayed informing you where the field is being used.

Chapter 14

VALIDATIONS

Validations is an option located on the Universals tab and in the Globals section of Program Details. Currently, the only available validation is a list of values. Lists of values are user created tables that hold values and labels that describe the values. Lists of values are assigned to field variables then used in other variables.

Lists of values are managed in Validations.

- Universal tab is where validations are created, edited, and deleted
- On the globals section of Program Details is where validation revisions are selected

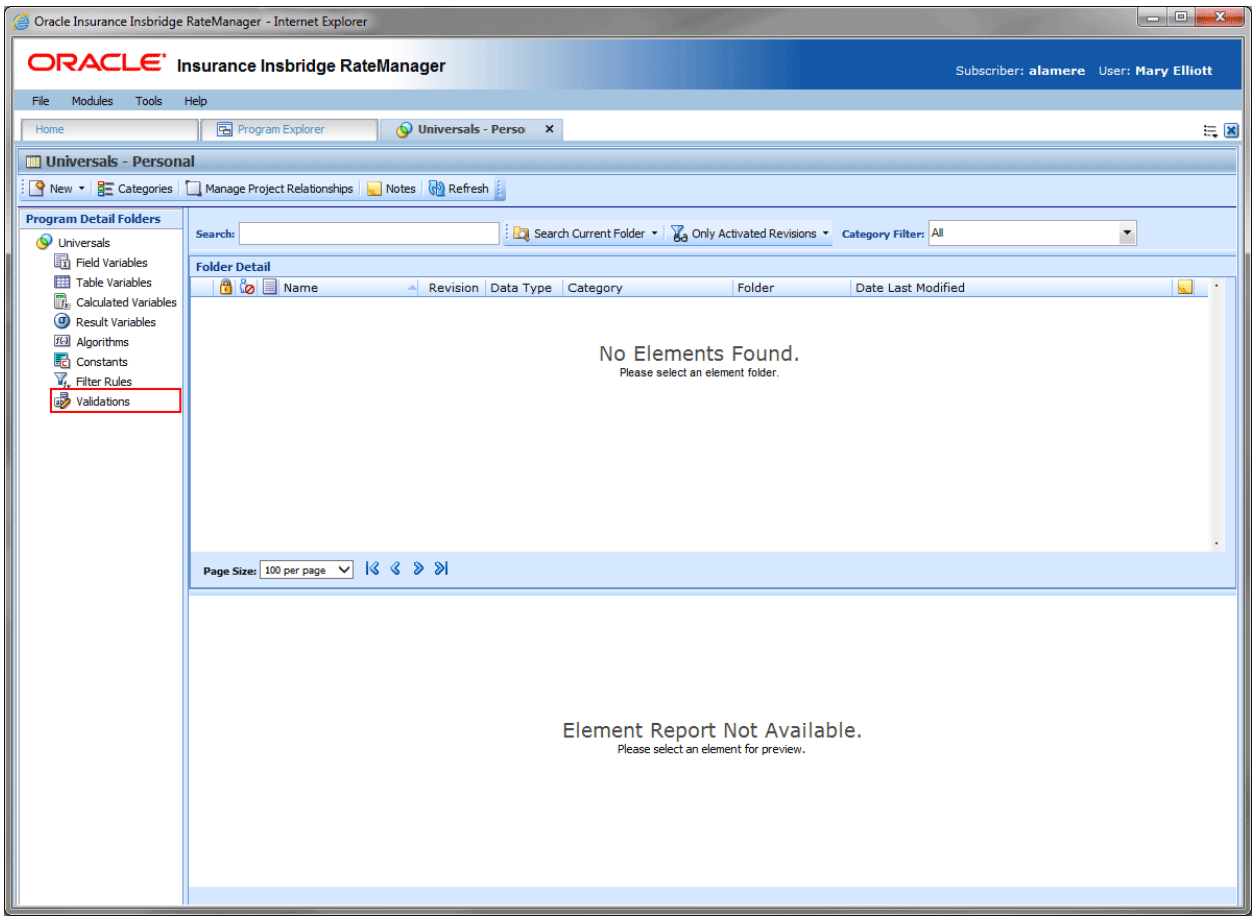


Figure 144 Selecting Validations

The purpose of a list of values is to provide the ability to select values based on their label/description.

For example, there are instances when a field value entered in the system may be a code or some other value that is not immediately recognizable. This may require a separate list to translate the code in to a more readable or recognizable form, creating an extra step and costing time. If the field has a list of values added to it, then you can click the list and see the value and a description. This can make editing easier and less time consuming.

Navigating to Validations

Validations are not an option on the Program Explorer tab.

1. Modules→Program Explorer to open the Program Explorer tab.
2. Select the domain type→domain→project→universals where you want to manage a validation.
3. On the universals tab, select Validations from the Program Detail Folders column.

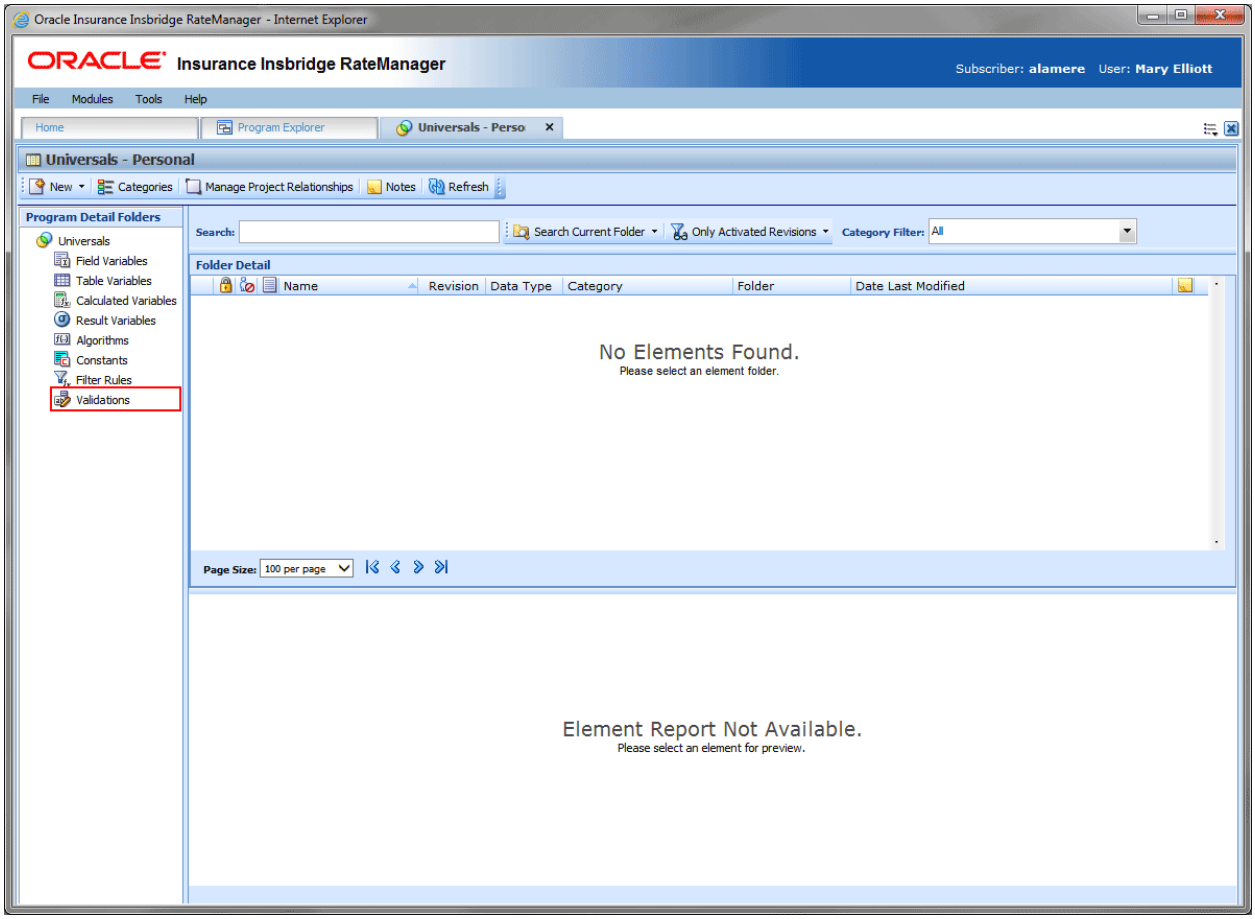


Figure 145 Validations Listing

VALIDATIONS FUNCTIONALITY

Functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any validation. The functions you can perform depend on the status of the validation. Locked validations have fewer options. There are icons that can tell you at a glance what the status of the validation is.

Icon	What it Indicates
	Validations. Validations are project specific and can be created at the universal level only. All validations are displayed on Program Details→Validations. The same validations listing also is

	available on the Project→Universals tab. The options new, edit, copy and delete are available only on universal tab. Revisions can be set on the Program Details tab only.
	Locked Validations. Locked validations are not open for editing. A lock icon indicates that this validation is locked by a program lock. If you have to edit a program locked validation, you must return to the locked program and unlock it. This unlocks the elements and changes can be made.
	In-Use. Validations that are in use are currently being worked on by another user in the system are not open for editing, copying, or deleting. The validation will be available when the user exits the validation.
	Advanced Option. This validation has a wildcard applied to it.
	Filter Rule. This validation has a filter rule applied to it.

VALIDATIONS DETAILS

The screen details are the same for the Universal tab and program details. For more on screen detail descriptions please see Program Details→Screen Details.

Validations Setup

There are additional setups that need to be entered in RateManager before users can start working with validations. You must have admin permissions to perform these tasks. If you cannot access validation, please contact your system administrator to setup permissions.

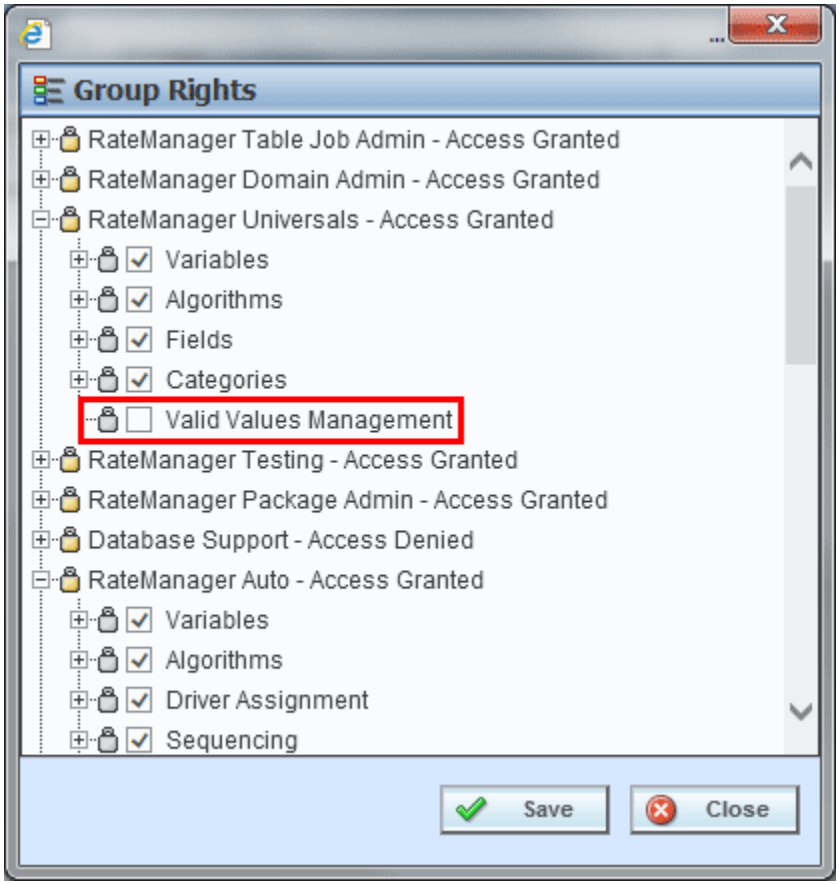
There are two set of permissions that can be applied;

- Permissions to create and edit validations
- Permissions to set active revision

Separate permissions keep the validations from being edited unintentionally but allows users to select the validation revision that is needed.

Permissions to Create and Edit Validations

To create, edit, and delete validations, permissions must be assigned at the group level. These permissions apply to all users assigned to the group. These are universal permissions and allow users to create, edit and delete validations in any domain.

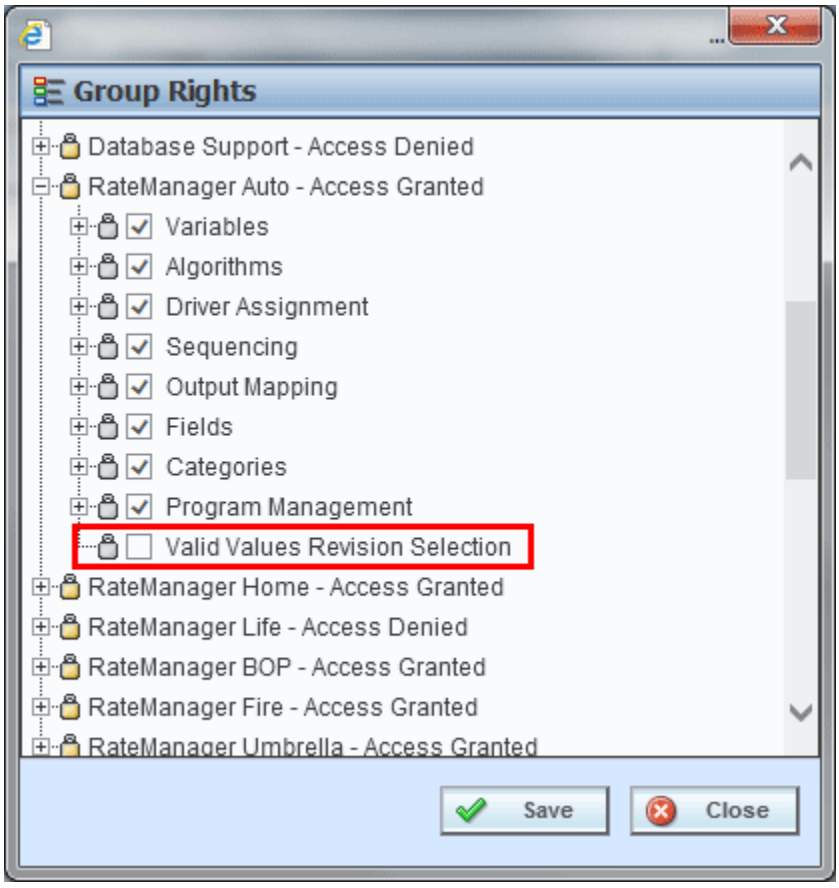


Admin permissions are required to edit Group Rights.

Expand RateManager Universals and check Valid Values Management. Save your entry. Users may need to reload to see the change.

Permissions to Set Active Revision

To select active revisions, permissions must be assigned at the product level. These permissions apply to all users assigned to the group. These are product level permissions and allow users to set active revisions in the product regardless of domain.



Admin permissions are required to edit Group Rights.

Expand RateManager Product and check Valid Values Revision Selection. Save your entry. Users may need to reload to see the change.

Valid Values Revision Selection must be selected under every product that the group has permissions to.

If valid value management rights are not granted, users can view valid values but cannot create, edit or delete valid values. If valid value revision selection rights are not granted, users will be able to view revisions but not set the active revision.

List of Values Tables

Prior to creating a list of values, a table must be created that contains the value and the label/description of the value.

List of values uses a two column table that contains a listing of values and labels. Values are the valid values and labels are the descriptions of the value. For example, if a ZIP code is submitted, you may not know what city that ZIP code represents. By adding a list of values to the field, you can toggle between views to see the value and the label.

The data type of the value must match the field variable where the list of values is to be assigned. The label can be any data type.

List of values can be imported in to the list of values in tab delimited or MS Excel format. List of values also can be created manually in RateManager.

	A	B
1	ZIP Code	City
2	00601	Adjuntas PR
3	00603	Aguadilla PR
4	00606	Maricao PR
5	00616	Bajadero PR
6	00638	Ciales PR
7	00641	Utuado PR
8	00647	Ensenada PR
9	00653	Guanica PR
10	00669	Lares PR
11	00670	Las Marias PR
12	00682	Mayaguez PR
13	00685	San Sebastian PR
14	00715	Mercedita PR
15	00720	Orocovis PR

Tables used by the list of values function like a table variable or data source. Date can be imported or exported in tab delimited or MS Excel format. Wildcards, advanced options and interpolation can be added.

USING A LIST OF VALUES

A list of values is a multi-step process.

1. The list of values is created in Validations. This is a 2 column table with the value and the label/description.

2. The list of values is added to a field variable. The data type of the field must be the same as the data type of the value used by the list of values.

3. The field is added to a variable. The field has an icon that indicates a list of values is associated with the field.

List of Values Rules

List of values are used in the design environment only. Adding a list of values to a field can be helpful when editing a large table, especially if codes are used and frequent lookups are done to see the code values.

- Only one list of values can be assigned to a field.
- The date format for list of values is determined by the first active product in the project.
- When used in a variable, the list of values dialogue box contains the first 25 values that match the search criteria. If no search criteria are entered, the first 25 values found on the table are returned.
- When used in a variable, the list of values dialogue box searches on labels, not values.
- The default view for the edit table data screen is value.
- When used in a variable, changing the show options changes all the fields with a list of values.
- List of values are used in design times definitions only and are not packaged with the program or used at run-time.
- List of values can be removed from the field at any time. If the program is packaged, the validation is removed from the field the next time the program is packaged.

CREATING A LIST OF VALUES

If permissions have been granted, you can create, edit or delete lists of values.

You will need to know the data type of the field where you want to place the list of values. The data type of the value must match the data type of the field.

Begin by accessing the validation option on the universal tab in the project where you want to create a list of values.

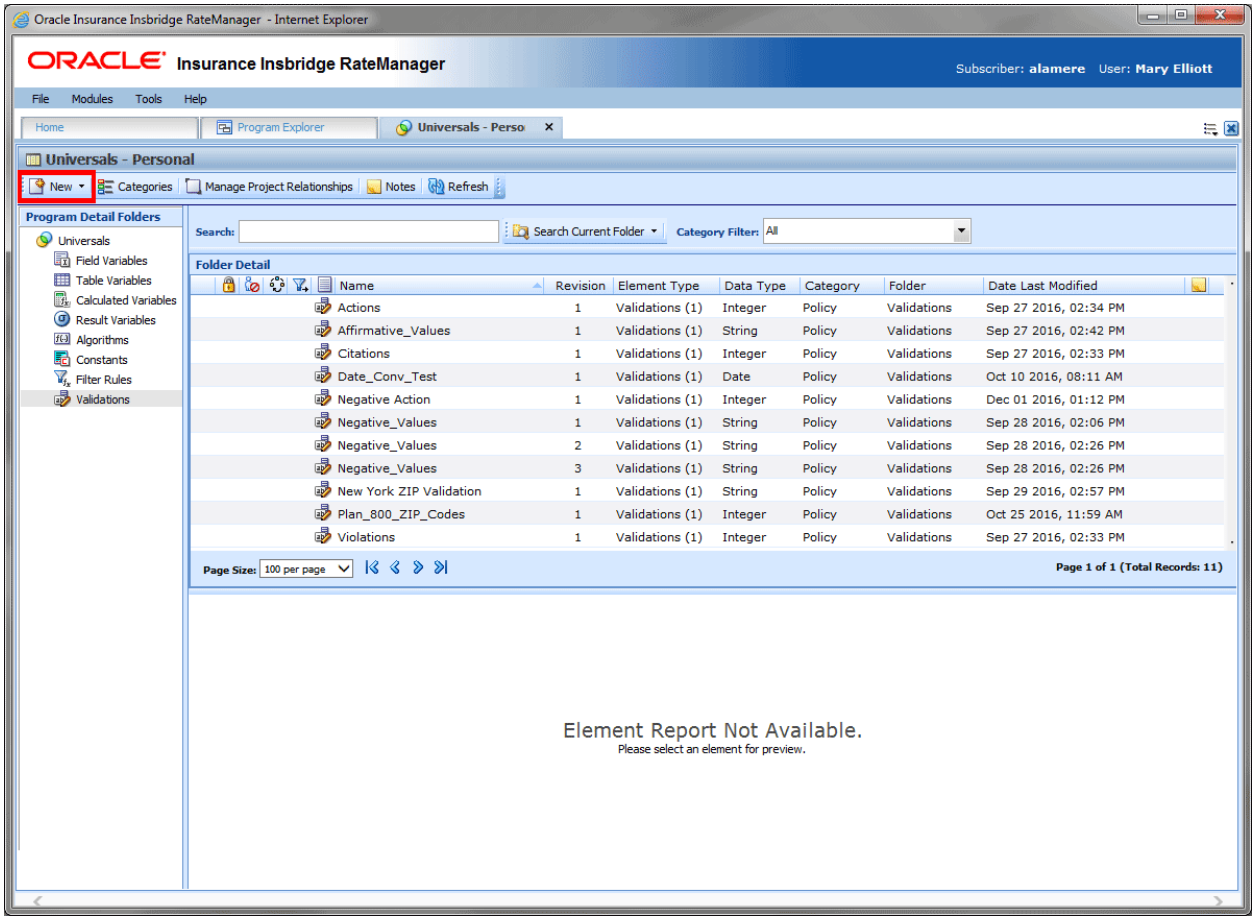


Figure 146 Creating a List of Values

1. Select Validations and click New.

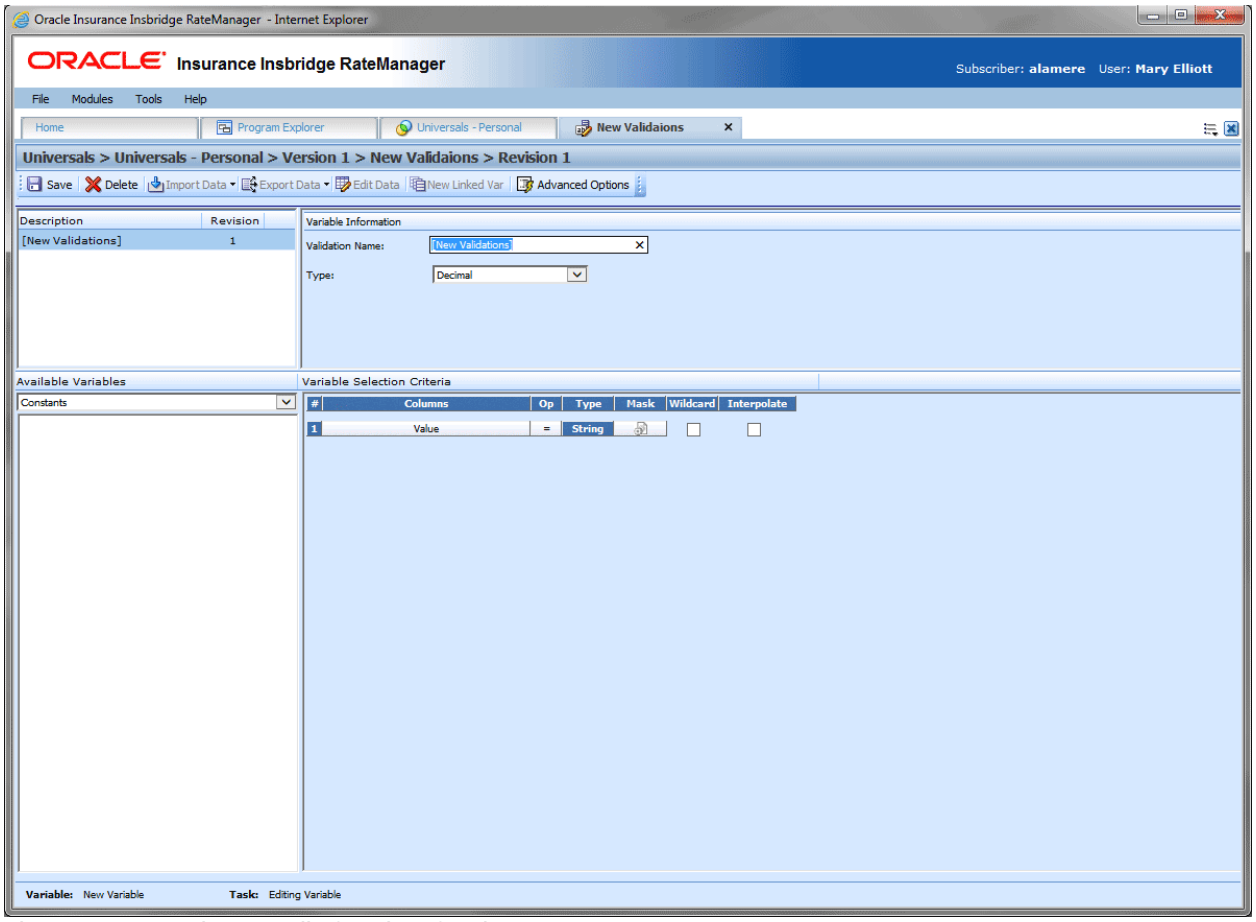


Figure 147 Entering Details for List of Values

2. Enter a validation name. This is the label or description for the field variable value. The label can be any data type.

3. Values are the list of values. The data type of the value must match the field variable where the list of values is to be assigned.

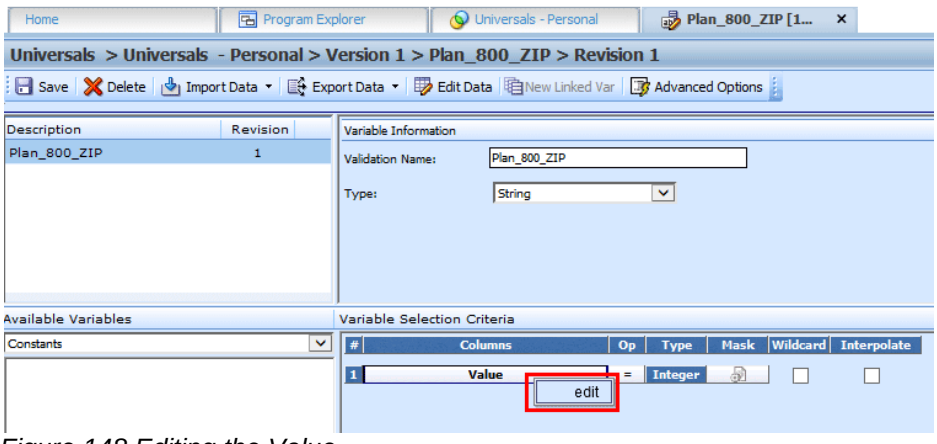


Figure 148 Editing the Value

4. To change the data type of the value, right click, select edit and then select the data type needed.

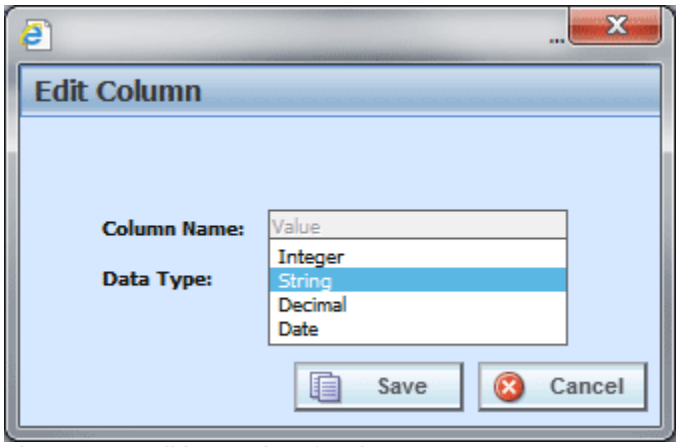


Figure 149 Editing a List of Values

5. If needed add any options. List of values can have other options added, such as wildcards, interpolations or advanced options.

6. Click Save.

7. Import the list of values. This is the table with the value to be entered and the label or description of the value. Data can be imported in tab delimited or MS Excel format. Data can also be entered manually. List of values tables function like table variables or data sources.

8. After the data has been imported, save.

ADDING A LIST OF VALUES TO A FIELD

The second step is to add to a field variable. When the validation is complete, it is ready to be added to a field variable. List of values can be added to a new or existing field.

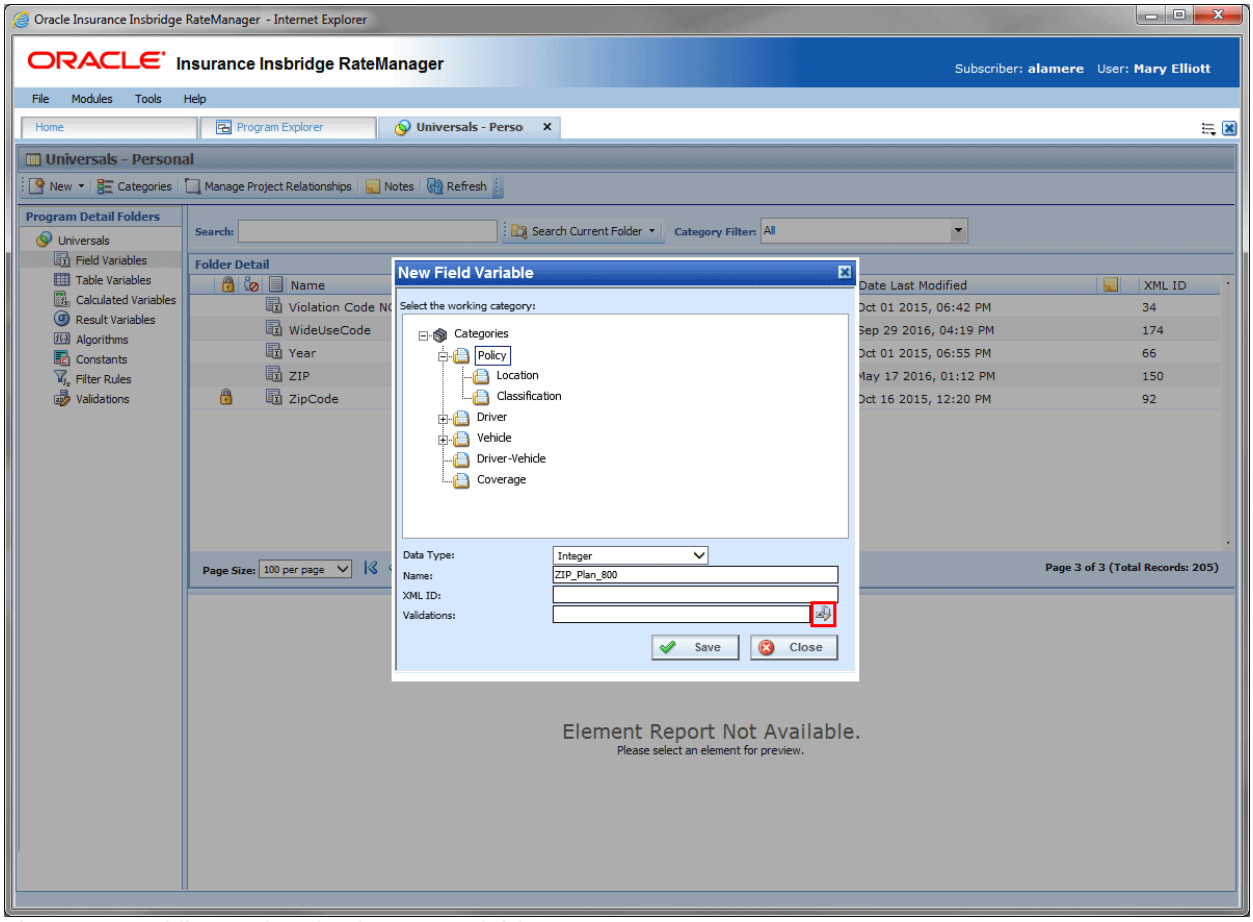


Figure 150 Adding a List of values to a Field

1. Select the field where you want to add a list of values or create a new field.

2. Click the validation icon at the end of the validation row.

3. A list of values with the same data type is presented. Only lists of values with the same data type are shown. If the list of values you need is not displayed, check the data type. You can search for the list of value you need by name. The list also can be sorted.

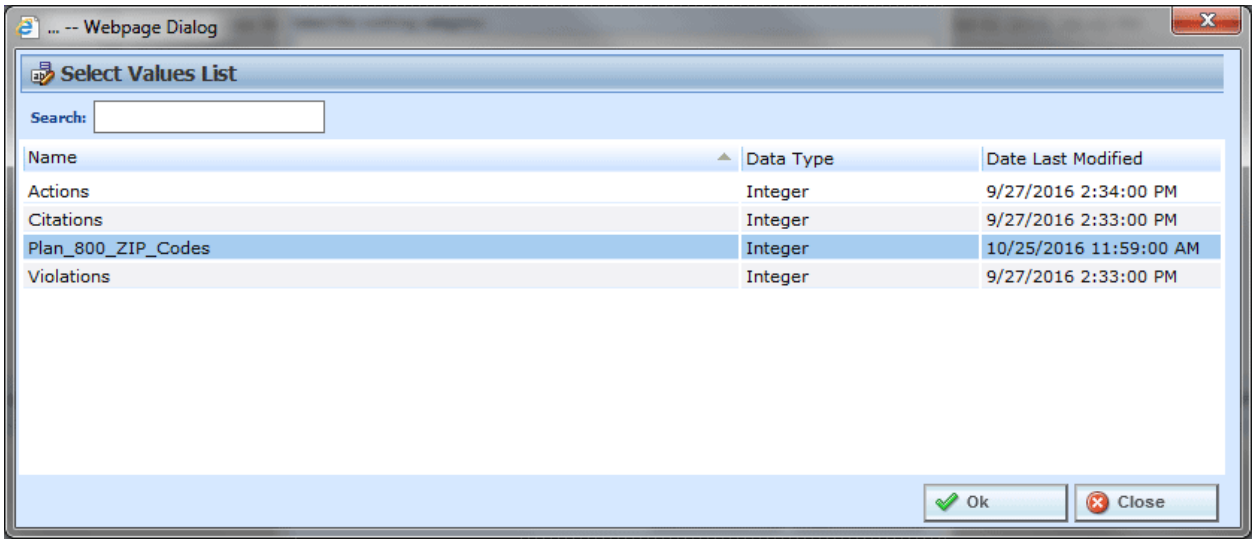


Figure 151 Selecting a List of Values

4. Select the list of values you need and click OK. Only one list of values can be assigned to a field.

5. Save your work.

6. After the list of values has been assigned, the field variable icon changes slightly, indicating a list of values has been assigned. The field can now be used in other variables.

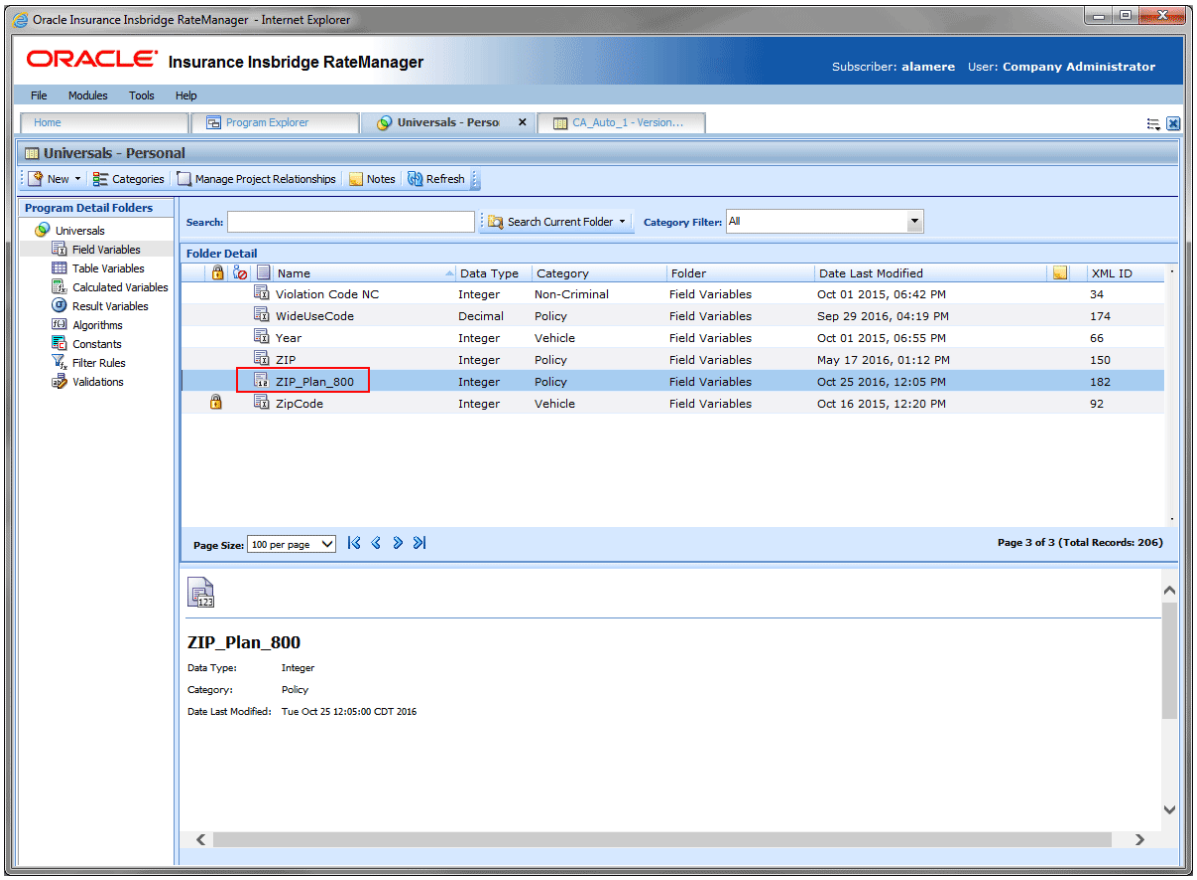


Figure 152 Field with List of Values Added

ADDING A FIELD TO A VARIABLE

The final step is to add the field variable with the list of values to a variable. The field can be used on table variables, data sources, calculated variables, algorithms and drivers assignments. In this example, a field with a list of values is added to a table variable.

Fields with a list of values are designated with a green check mark. One or more fields with a list of values can be used in a variable and the fields can be in any order.

1. Create a new variable or edit an existing one.

2. Select the field with the list of values. When a field with a list of values is added, the list of values that was assigned to the field is displayed at the end of the row. This is information only.

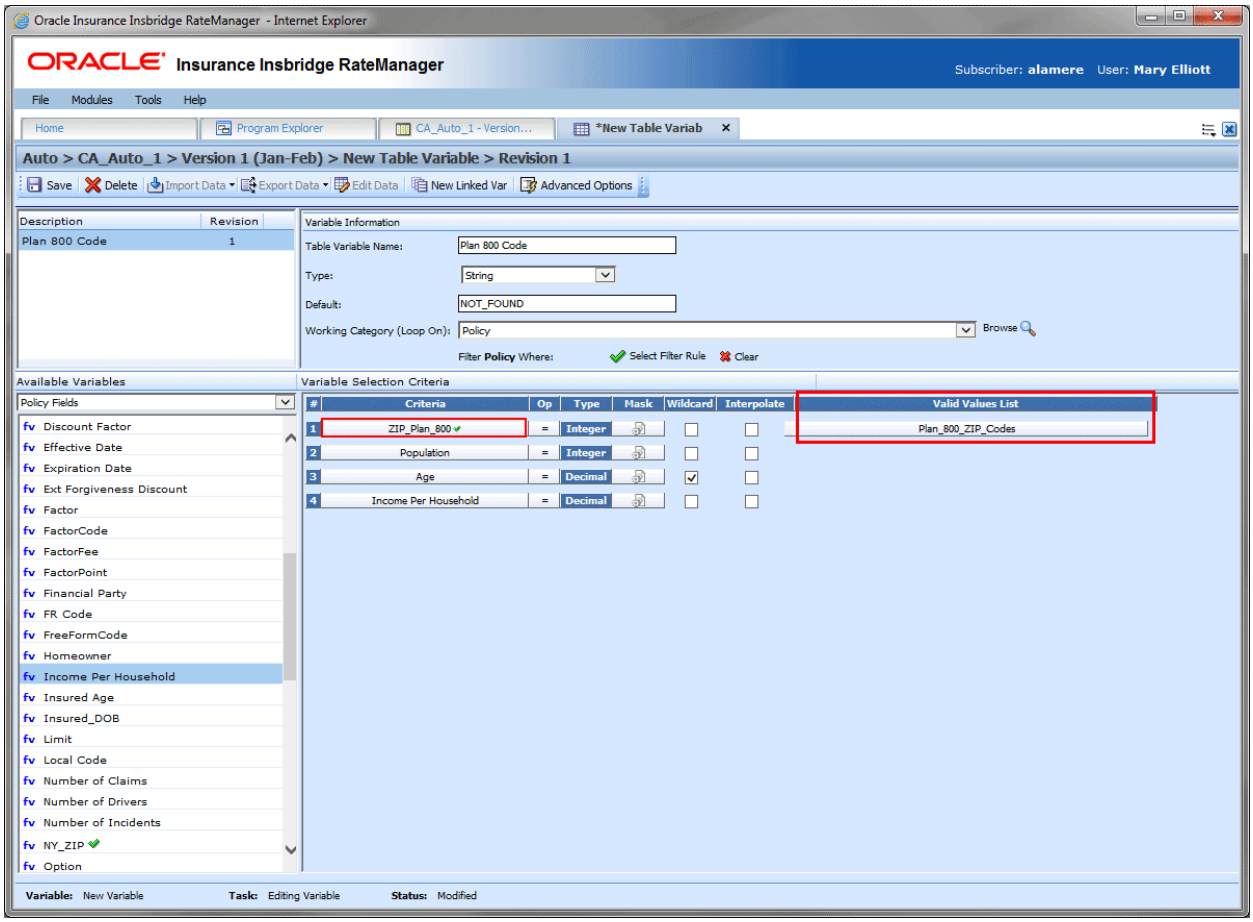


Figure 153 Table Variable with List of Values Field

3. Save your entry. Data can now be imported in to the table.

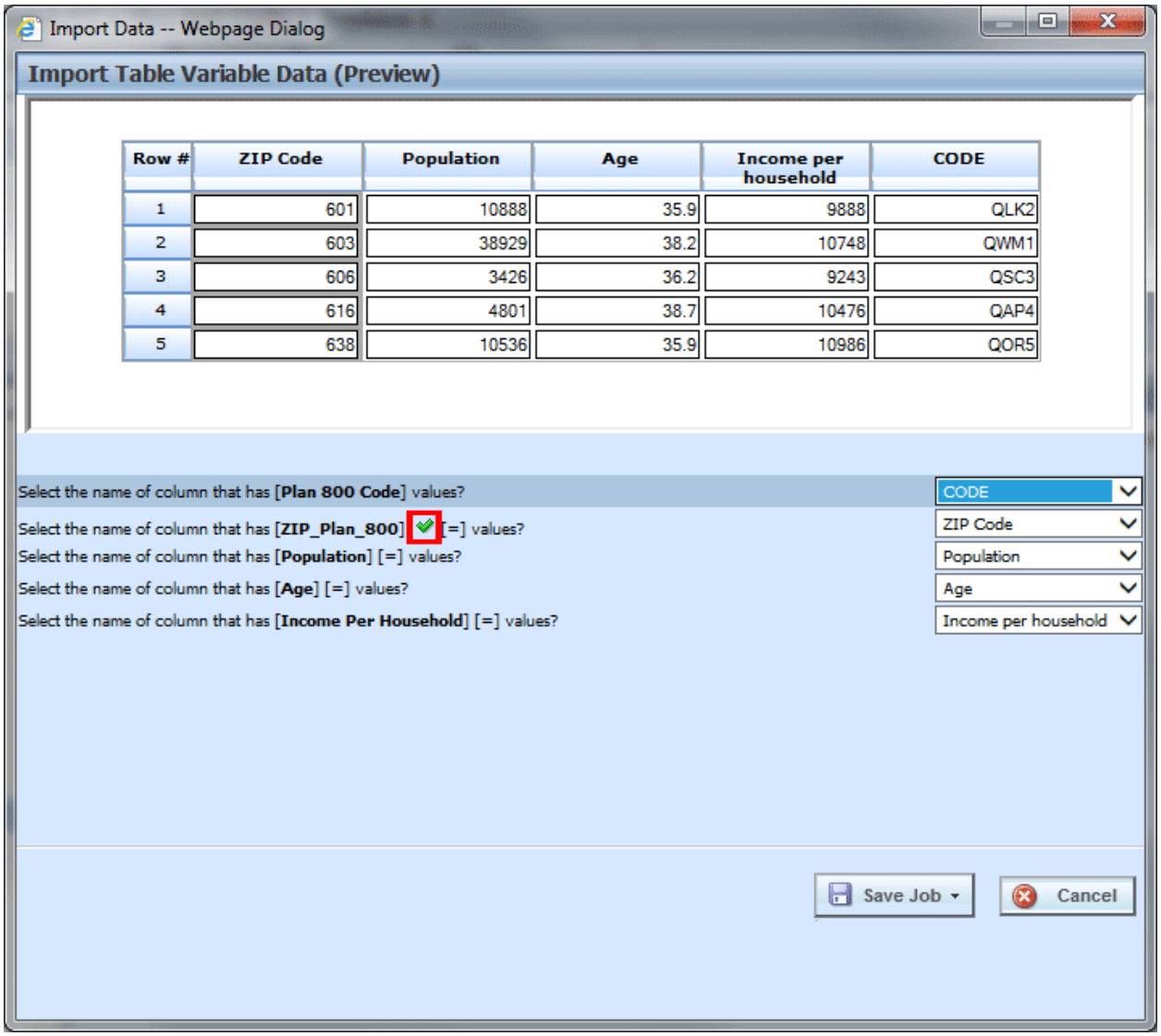


Figure 154 Importing Data with List of Values Field

4. When data is imported in tab delimited or MS Excel format, a green checkmark is placed next to the column name where the field has a list of values. The data to be imported is the value.

Editing Table Data with List of Values Fields

On the table data edit screen, lists of values are indicated with a green checkmark in the column header. The list of values icon is next to the cell.

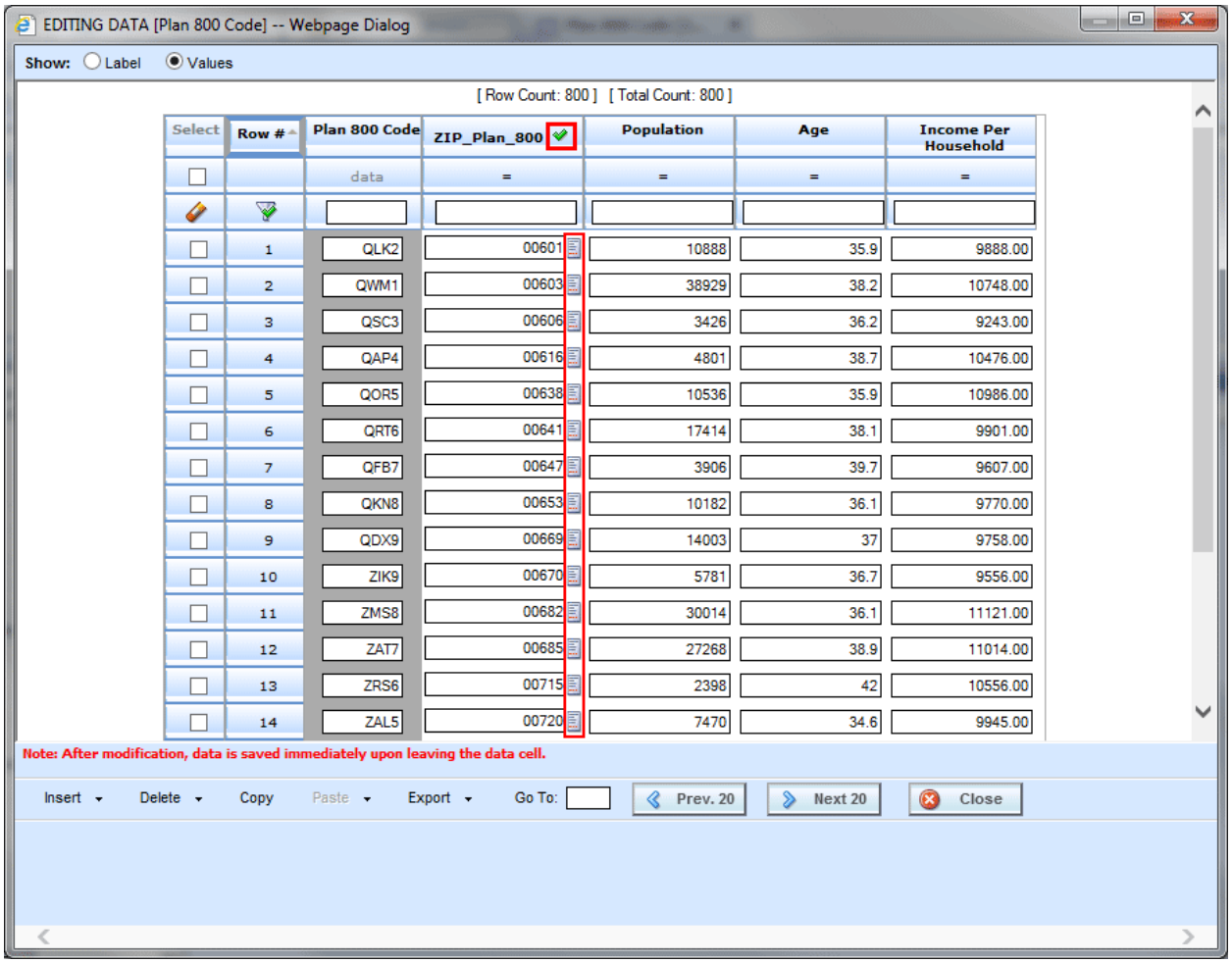


Figure 155 Editing Data with a List of Values Field

Using a List of Values

When a list of values icon is clicked, the valid values for the field are listed. The first 25 values are listed regardless of where you clicked the icon. If you click the list of values icon on row 600, the first 25 values in the table are displayed. If you need a value beyond the first 25, you can search for it.

Searches are done on the label. To do a search, enter your criteria in the search field and matches will populate the listing. Up to 25 labels that match the search criteria are displayed.

Cells on the table data can be updated by selecting a label/value and clicking Use Value. Changes are immediate and cannot be undone.

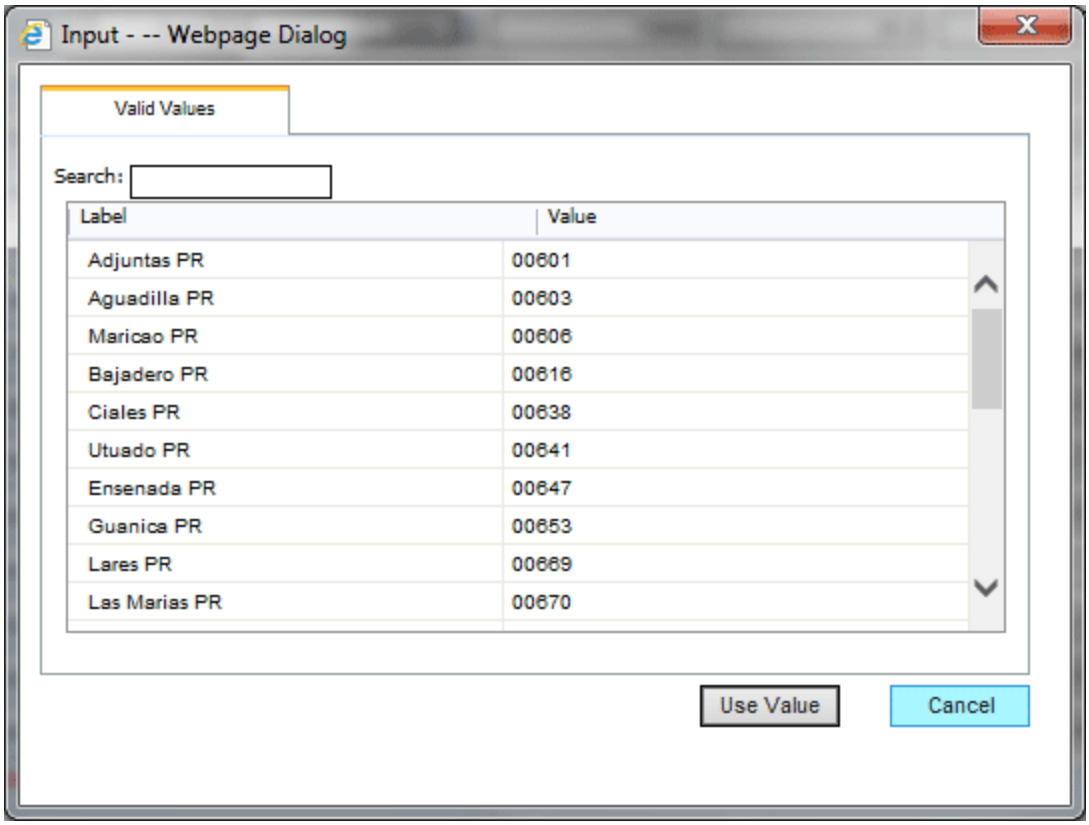


Figure 156 Valid Values for Table Data

Changing the View

When a field with a list of values is included in a table variable, you can choose to view the field by the value or the label. The default view is the value.

Changing the Show option changes the display for the currently selected screen, unless a filter is in place. When a filter is in place, the show option will stay for the rows returned.

When the Label option is selected, the label is displayed in the cell.

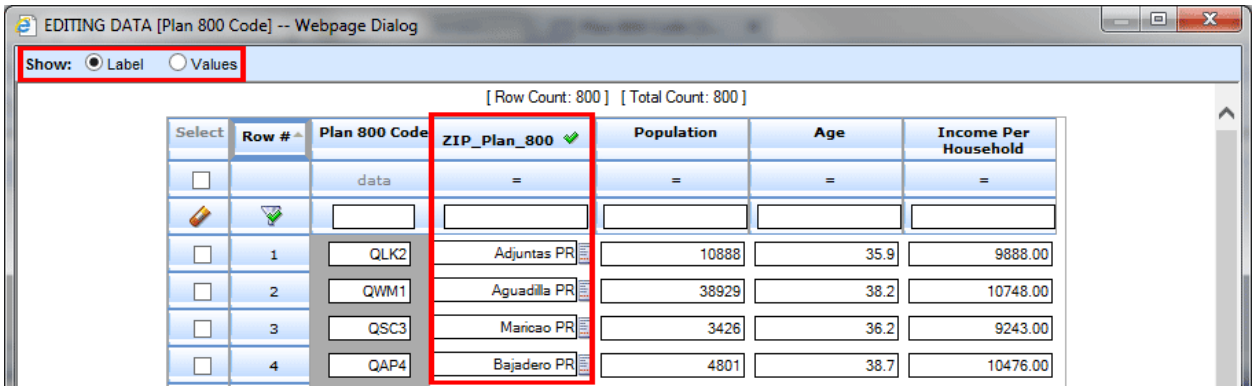


Figure 157 Selecting to View Table Date by Label

Working with List of Values Data in Test Case Editor

The green check make indicates that the field has a list of values assigned to it. At the end of the entry is the list of values icon.

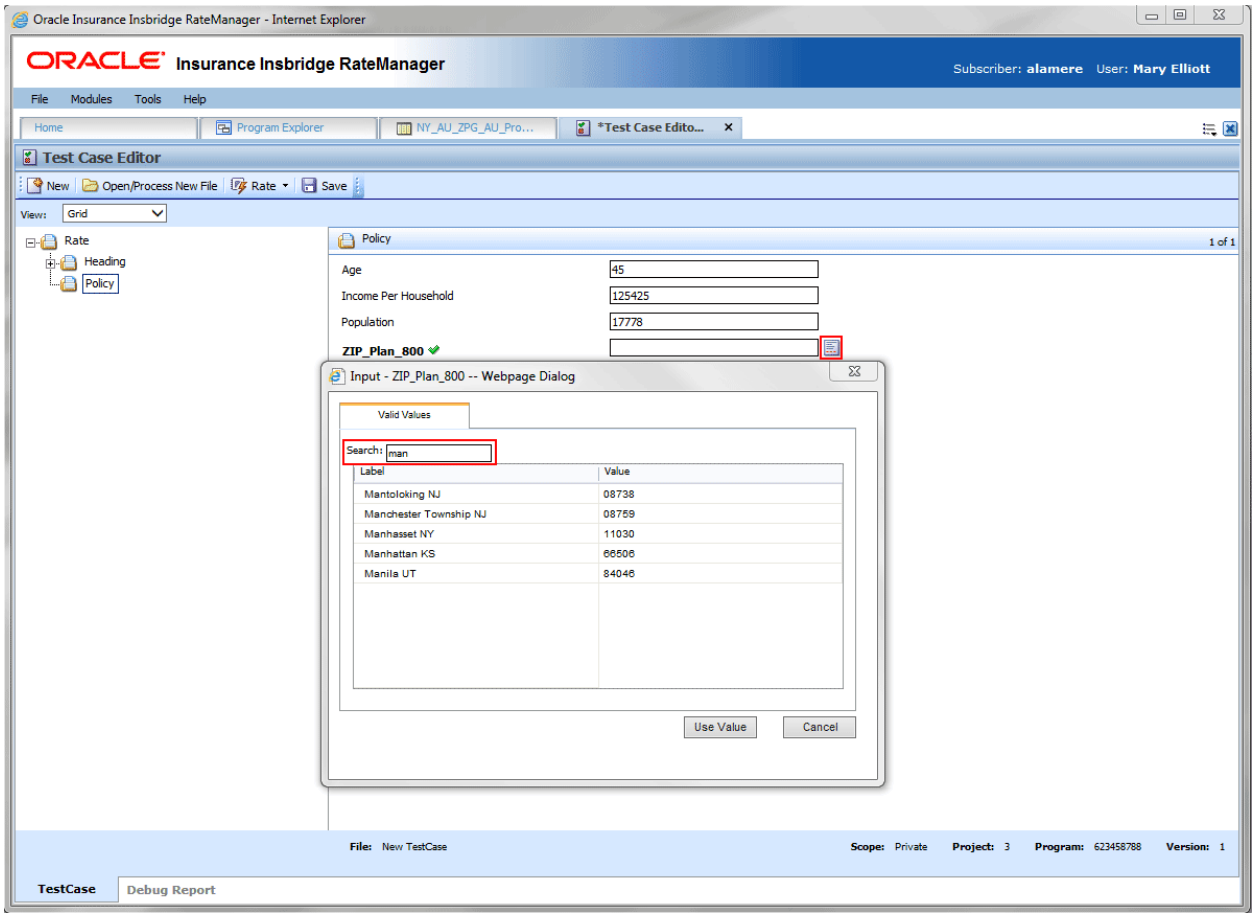


Figure 158 List of Values in Test Case Editor

For Test Case Editor only the first 25 values are returned when the valid values icon is clicked.

When a list of values icon is clicked, the valid values for the field are listed. The first 25 values are listed. Select the value you need. If you need a value beyond the first 25, you can search for it.

Searches are done on the label. To do a search, enter your criteria in the search field and matches will populate the listing. Up to 25 labels that match the search criteria are displayed.

Selecting a label/value and clicking Use Value populates the entry with the selected value.

On the debug report, the field with a list of values assigned to it is shown with a green asterisk.

Editing a List of Values

List of values can be edited on the universal tab only. Unlocked list of values can be edited at any time. If the list of values is locked because the program is locked, you must return to the program and unlock it. Once unlocked, changes can be made. If the list of values is locked because it came from a template, you cannot unlock it.

List of values allows for name changes and data type changes. Data type changes are enforced when you edit the data in the table or assign to a field.

NOTE: *If a change is made to a list of values, the change affects all programs that use that field.*

1. Either double click the list of values you want to edit or right click and select Edit. This opens the Edit screen.

2. Make any changes to the Data Type and Name. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Click Save to save your changes.

4. If you need to edit the data, click the Edit Data option in the top menu.

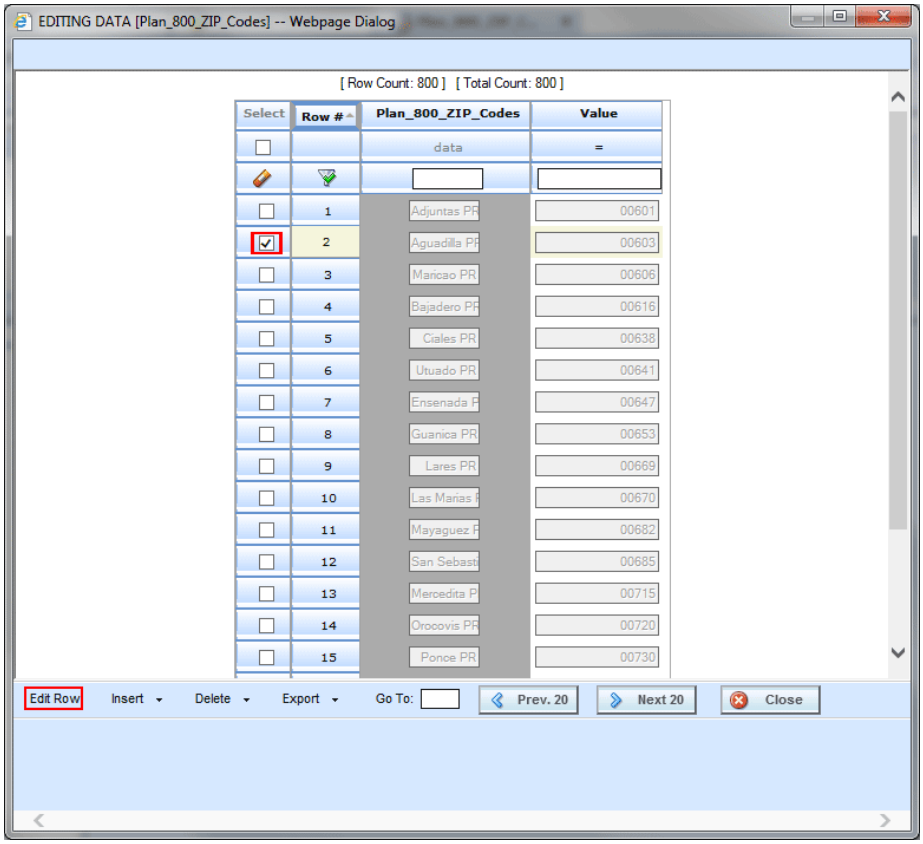


Figure 159 Editing a List of Values Data Table

5. List of values data tables are edited like Data Sources. Select the row where you want to edit and click Edit Row.

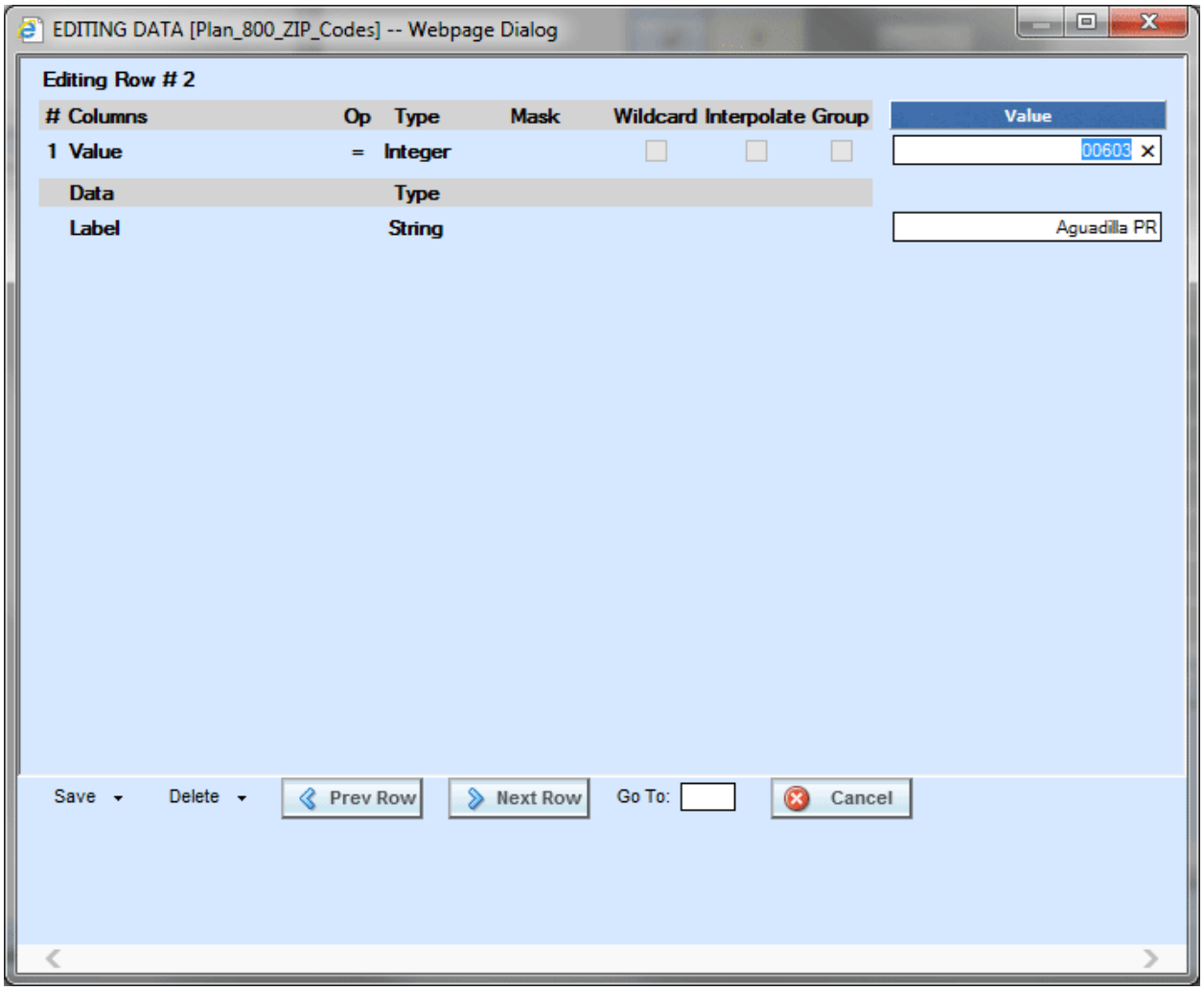


Figure 160 Editing a Row on the List of Values Data Table

6. Make your changes to the row and save and close, save and next, or save and new. Navigation can be done using the Next and Previous row options, also the Go To option. For more on editing, please see Editing Data Sources Data.

7. After editing the data, close the edit data window. Click Save to save your changes.

Table data can be re-imported if needed.

Copying a List of Values

Lists of values can be copied at any time. The list of values is copied with the exact same data types. There are two copy options; copy definition only, or copy definition and all data. List of values can only be copied at the universal level on the universal tab. To make a change to a copied list of values, you must go into the edit screen. Names cannot be duplicated.

1. Right click the list of values you want to copy. Select Copy. The Copy window is displayed.

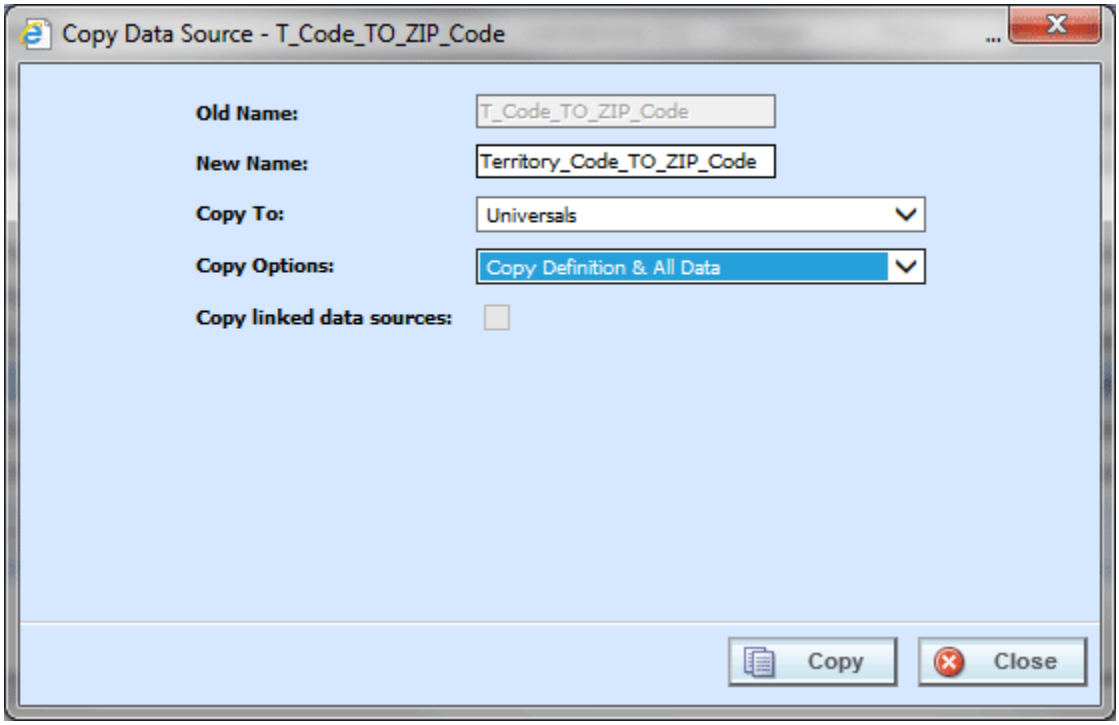


Figure 161 Copying List of Values

2. Enter in a New Name. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Select the copy option you want; definition only or definition and all data.

4. Click Copy to copy your list of values. The screen refreshes and the copied list of values is shown on the universal screen→validations listing.

Setting the Active Revision of a List of Values

Active revisions for list of values are selected from the globals area of Program Details. You must be in a program to select the active revision. List of values are not available on the globals tab. Revisions cannot be selected from the universals tab.

Set the active revision before assigning the list of values to a field.

1. Enter the Program Details tab of the program where you want to select the active revision.

2. Select Validations from the globals area.

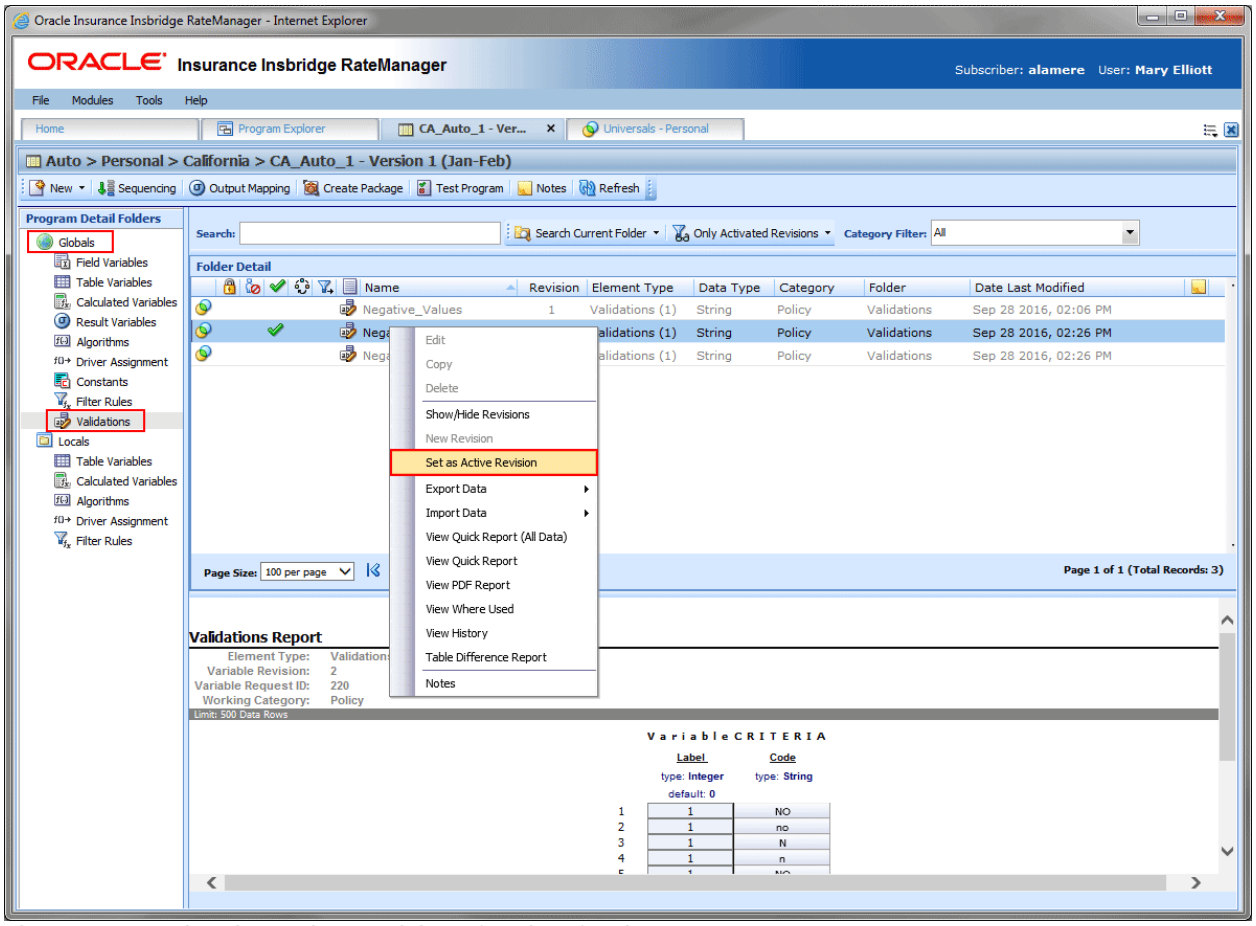


Figure 162 Setting the Active Revision of a List of Values

3. Select the list of values you need and click Show/Hide revisions.

4. Select the revision you want and right click.

5. Select Set as Active Revision.

6. A message is displayed indicating that you are about to change the active revision. Click OK to change the active revision. Click Cancel to exit the window. Changes are immediate.

DELETING A LIST OF VALUES

Any unlocked list of values can be deleted. List of values that are in use or locked cannot be deleted. If you try to delete a list of values that is unavailable for deletion, an error message is displayed informing you where the list of values is being used.

NOTE: *Take care when deleting Lists of Values, this action cannot be undone.*

1. A list of values can be deleted by right clicking the list of values and selecting **Delete**.

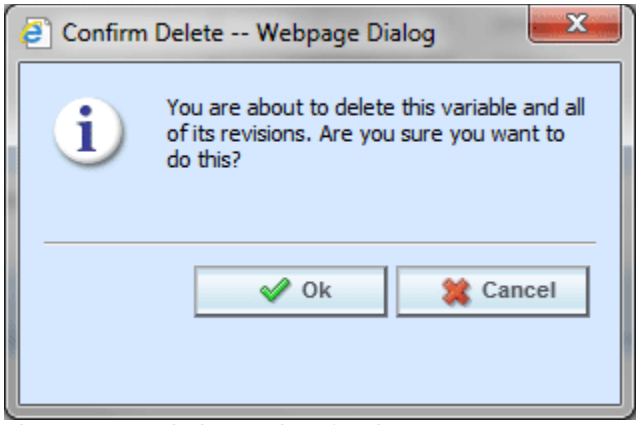


Figure 163 Deleting a List of Values

2. A confirmation message is displayed.

3. Click **OK** to delete or **Cancel** to cancel. If the list of values is being used, you cannot delete it. A message is displayed informing you where the list of values is being used.

RESULT VARIABLES

When a variable is created under a specific program, it is referred to as a local variable. When a variable is created that applies to all programs under the product, it is referred to as a global variable. When an element is created that can be used by all programs under the project, it is referred to as a universal element. Result variables can be created at the global level and at the universal level.

Results variables generally hold the result of an action and can be used at any point in the system. When calculating a premium, the final result is usually a result variable. Result variables can also be used as placeholders for the end result of an algorithm. This includes premiums, underwriting rule messages and policy specific information such as tier name selected, and fees.

It is recommended that result variables be created prior to algorithms and calculated variables so that these elements are in place for selection from the drop down listings when you need them in your calculation sequence.

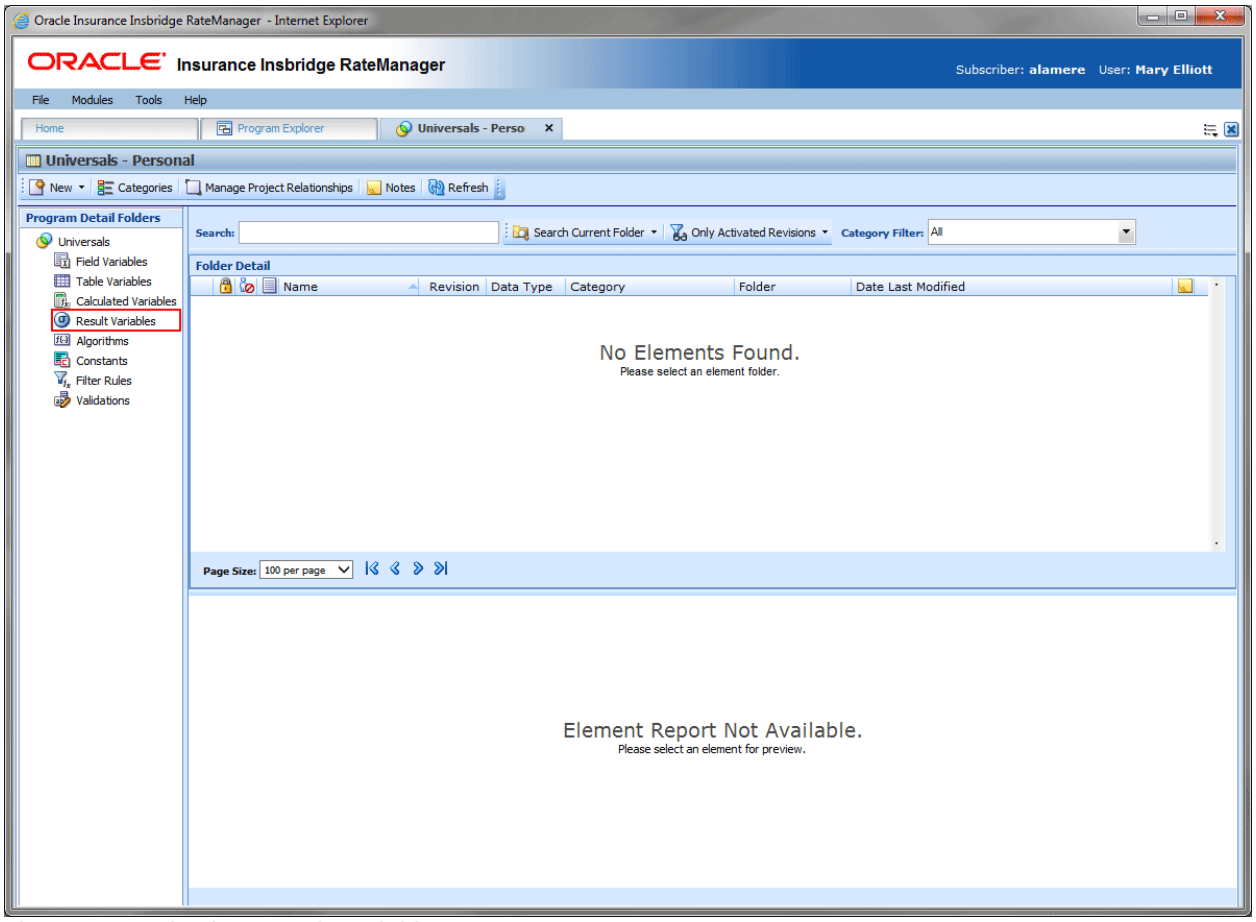


Figure 164 Selecting Result Variables

Result variables can be created by you or brought in from a template. Variables that are user created are open for editing, copying, and deletion if they are not locked. Locked result variables can be copied but not edited or deleted. Variables that were brought in from a template are locked at all times and cannot be unlocked. No editing or deleting is allowed. This is to ensure the integrity of the template. Template

variables can be copied. New user created result variables can be added to a template program. The new variables and copies are open for editing, copying, or deletion.

Navigating to Result Variables

Result Variables are not an option on the Program Explorer tab.

1. Modules→Program Explorer to open the Program Explorer tab.

2. Select the domain type→domain→project→product→folder→program→program version. You also can access Result Variables from the globals tab, domain type→domain→project→product→Globals.

3. **Open** the program version to get the Program Details tab. Result Variables are available from the Program Details Folders area on the left side of the tab.

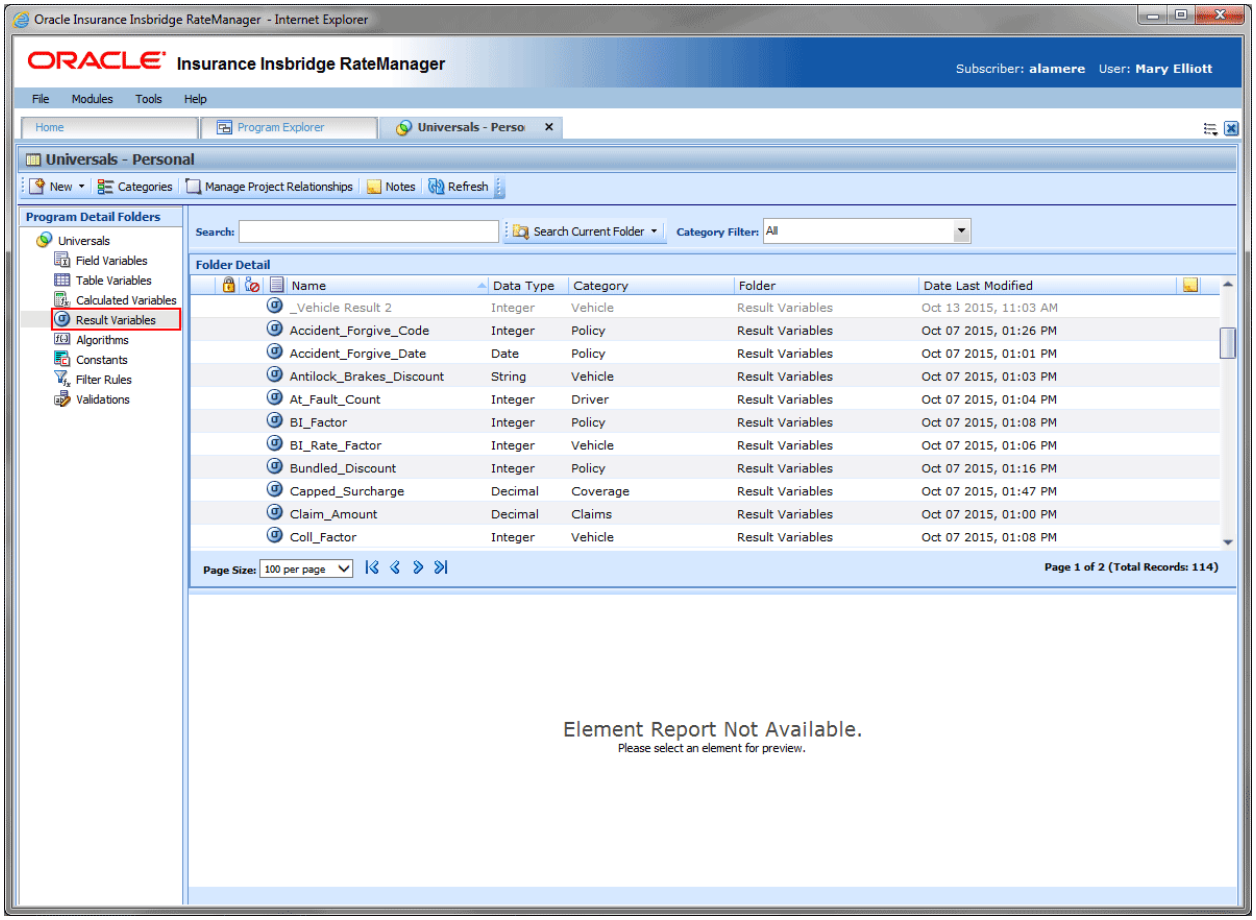


Figure 165 Result Variables Tab

RESULT VARIABLES FUNCTIONALITY

Functions can be accessed from the right click menu of any unlocked result variable. The functions you can perform depend on the status of the result variable. Locked result variables have fewer options. There are icons that can tell you at a glance what the status of the result variable is.

Icon	What it Indicates
	Result Variable. Result variables can be created at the global level or the universal level, from a program version, or from another program element. All result variables are displayed on the Program Details→Result. The same result variable listing is also available on the globals tab. The same options, new, edit, copy and delete are available on either tab.
	Universal. Indicates that the result variable was created at the universal level. The result variable is grayed out and can be edited, copied or deleted from the universal tab only.
	Locked Result Variable. Locked result variables are not open for editing. A lock icon indicates that this result variable is locked by a program lock and is not open for editing, or deleting. Program locked results can be copied. If the result variable is locked because the program is locked, you must return to the program and unlock it. This unlocks the result variable and changes can be made.
	Template Lock. If the result variable is locked because it came from a template, you cannot unlock the result variable. You can copy the result or create a new result variable for use in the program if necessary.
	In-Use. Result variables that are in use are currently being worked on by another user in the system are not open for editing, copying, or deleting. The result variable will be available when the user exits the result variable.

RESULT VARIABLES DETAILS

The screen details are the same for the Global Tab and Program Details. For more on screen detail descriptions please see Program Details→Screen Details.

Right Click Menu Options

To view right click menu options, highlight the result variable you want to view and right click. Unlocked result variables have six right click menu options:

- **Edit:** Allows for the result variable to be edited.
- **Copy:** Brings up the copy screen for the result.
- **Delete:** Removes the result variable from the list. You must have permissions to delete.
- **View Where Used:** Displays an information screen that details where the variable is being used. See View Where Used for more information.
- **View History:** Displays the Audit History report. A history of actions taken and provides a

detailed account of the action taken, who performed the action, the date the action happened, program information and any element information. This is an optional feature that must be enabled by the system administrator.

- **Notes:** Enter, edit and view notes for the result variable.

Template locked result variables have four options, Copy, View Where Used, View History and Notes.

Program locked result variables have three options, View Where Used, View History and Notes.

CREATING A RESULT VARIABLE

In order to create a new result variable, you must have your categories setup. See Categories for more information.

Result variables can be created inside of an algorithm using available categories. If you have to create a result variable using a new category, you must exit the algorithm and create the new category. Then you can create the result variable using the new category from either the Result Variable Tab or inside the algorithm.

Naming standards can be applied to system elements. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the element and for case enforcement to be applied to any newly created element or any existing elements that are being edited. If a naming standard is in place, the pattern is displayed in the element name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented for all elements.

1. On the Program Details tab, select Result Variables from the Program Detail Folders. Click **New** in the top bar menu. This opens the **New Result Variable** window.

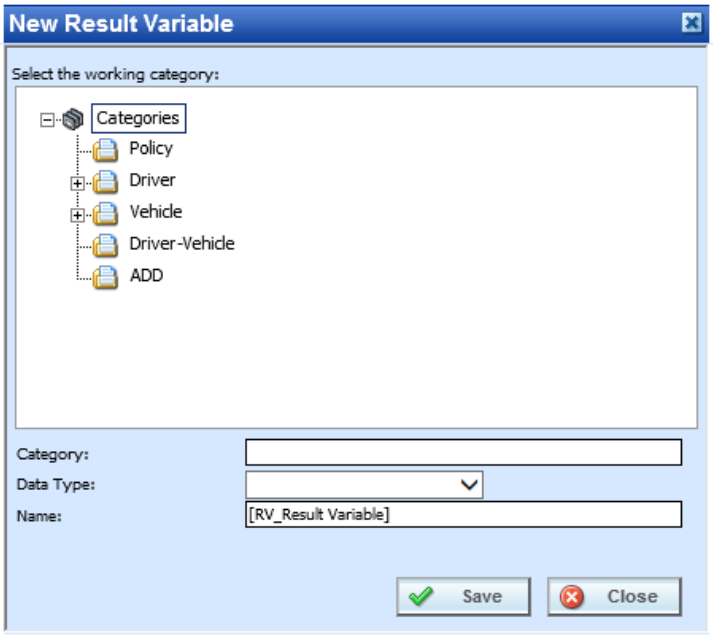


Figure 166 New Result Variable

2. Select the **Category Pathway**. The pathway is displayed in the Category field. On the tree view, the final destination where the new result variable will be placed will have a box around it.

3. Select the **Data Type** (date, decimal, integer or string) from the menu.

4. Enter a **Name** for the new result variable. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

5. Click **Save** when you are finished. The screen refreshes and the new result variable is displayed on the Program Details→Result Variable tab.

Adding a Result Variable from another Area

A new result variable can be created while entering or editing an algorithm without leaving the Edit tab. The category you can add the result variable to is determined by the Working Category selection. Any result variable created here can be edited on the Result Variable Tab, if needed.

If the new result variable needs to be in a new category, you must leave the Edit tab and create the category in the Program Explorer→Universals tab and then return to the Edit tab or the Program Details tab to create your result variable.

The result variable created inside another area is created in the globals area. If you need the result to be in the universal area, you must leave the Edit tab or the Program Details tab to create your result variable.

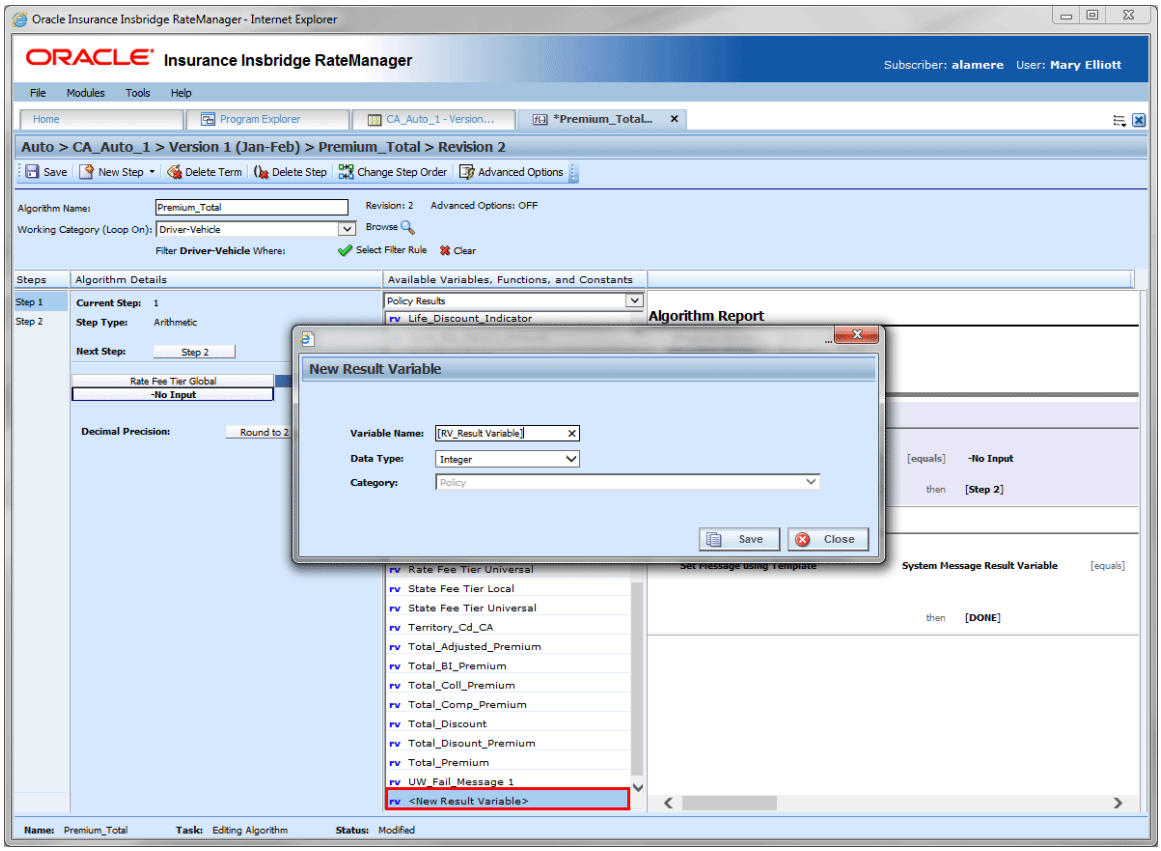


Figure 167 Adding a Result Variable

1. To create a new result variable, scroll down to the bottom of the category results list. Select **<New Result Variable>**. The New Result Variable window is displayed.

2. Enter in a **Variable Name**. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Select the **Data Type**.

4. Click **Save** to save your entry. The new result is now available for use.

Editing a Result Variable

Unlocked result variables can be edited at any time. To edit a locked result variable, you must return to the locked program and unlock it. Variables that were brought in from a template are locked at all times and cannot be unlocked. A new result variable can be created if necessary.

NOTE: *If a change is made to a result variable, the change affects all programs that use that result variable.*

1. Double click the result variable you want to edit or right click it and select **Edit**. This opens the **Edit Result Variable** window.

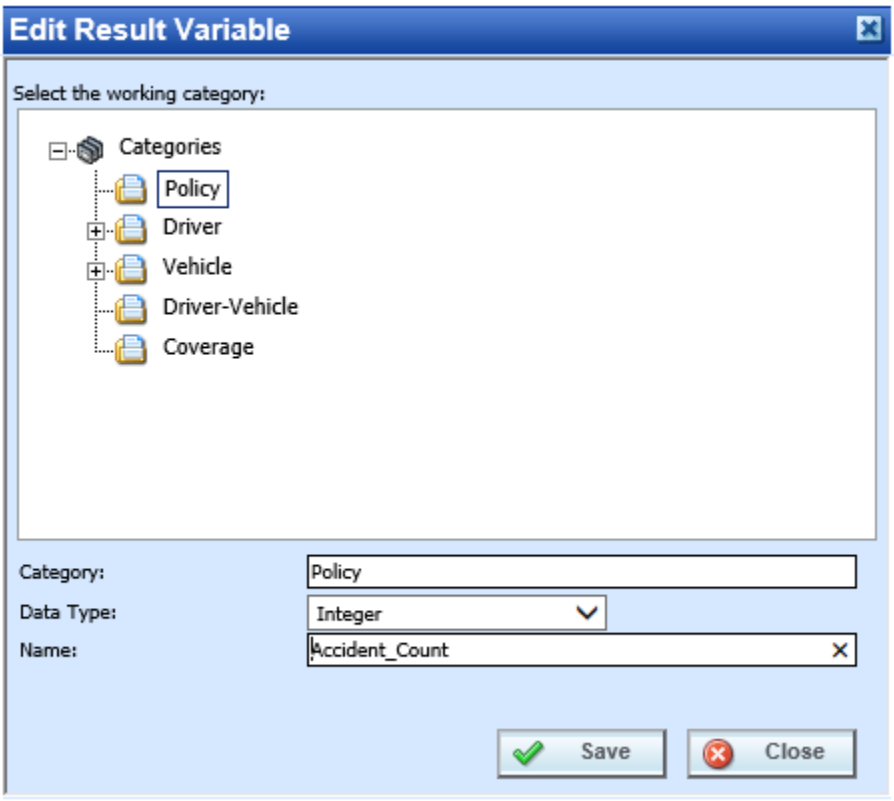


Figure 168 Editing Result Variable

2. To change the **Category**, double click the category where you want the variable to fall under. The pathway is displayed in the Category field. On the tree view, the final destination where the

variable will be placed has a box around it.

3. Make any changes to the **Data Type** and **Name**. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

4. Click **Save** to save your changes.

Copy a Result Variable

Any unlocked result variable can be copied at any time. The result variable is copied with the exact same data type and category. The copied variable is open for editing. To make a change to the variable, you must go into the edit screen.

A copied result variable requires a new name. No other information is required.

Copying a Result Variable

1. Right click the result variable you want to copy. Select **Copy**. The Copy Result Variable window is displayed.

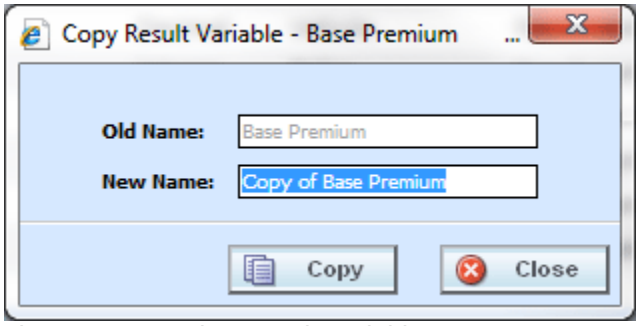


Figure 169 Copying Result Variable

2. Enter in a **New Name**. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Click **Copy** to copy your result variable. The screen refreshes and the copied result variable is displayed.

Deleting a Result Variable

Any unlocked result variable that is not in use can be deleted. Result variables that are in use, locked, or template generated cannot be deleted. If you try to delete a result variable that is unavailable for deletion, an error message is displayed informing you where the result variable is being used.

NOTE: *Take care when deleting result variables, this action cannot be undone.*

1. Right click the result variable you want to delete and select **Delete** from the menu.

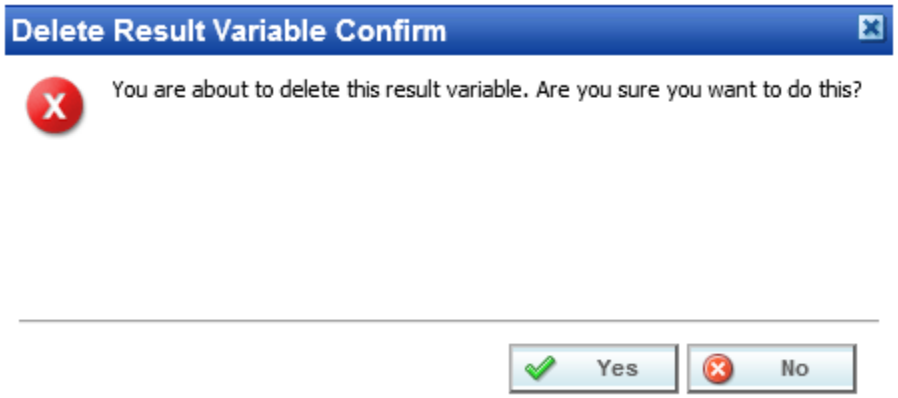


Figure 170 Deleting a Result Variable

2. A confirmation message is displayed. Click **Yes** to delete or **No** to cancel. If the result variable is being used, you cannot delete it. A message is displayed informing you where the variable is being used.

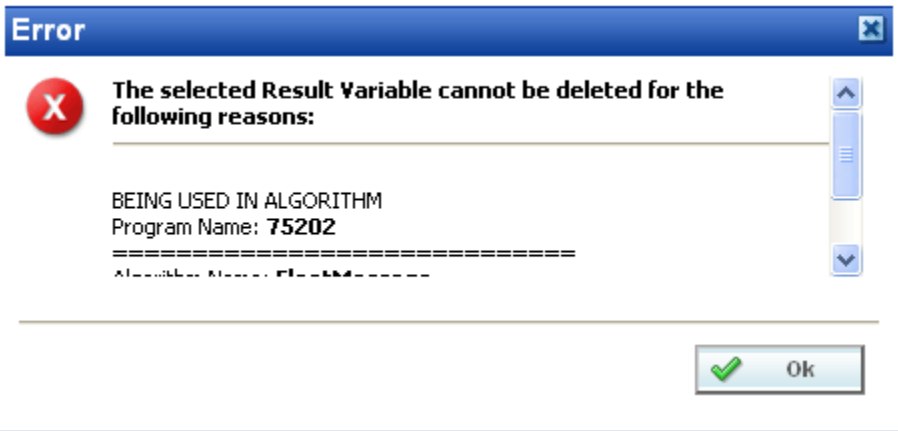


Figure 171 Result Variable Error Message

Chapter 16

TABLE VARIABLES

Variables are placeholders that get filled in during rating, based on fields and other criteria. In RateManager, a table variable is a table based or look-up variable. Any data that can be quantified in a table can be loaded into RateManager as a table variable. Common tables in a manual include territory codes, deductible factors and base rates. Also, rating data, such as credits and surcharges, can be defined in a table. It is usually advantageous to have as much rating data in a table as possible. Tables can be maintained and updated relatively quickly.

Table variables can be created and viewed at three levels:

Universal: Universal variables are created and managed at the universal level and are project specific. Universal variables can be managed only from the Universal tab. Universal variables can be viewed on the Program Details tab but no edits, copies, or deletions can be done.

Global: Global variables are created and managed at the global level and are product specific. Global variables can be accessed from the Global tab or from the Program Details tab.

Local: Local variables are created and managed at the program version level and are specific to the program. Local variables can be accessed from the Program Details tab.

There are a few differences between universals, globals and locals. One difference between them is that universal variables are available to all programs in the project. Global variables are available to all programs within the product while program level variables are available only to the program where they were created. Another is that universals can use only universal elements; global variables can use universal and global elements. Local variables can use universal, global, or local elements.

Table variables can be template generated or created by you. Variables that are user created are open for editing, copying, creating new revisions and deletion if they are not locked.

Template generated table variables are locked and cannot be unlocked. No editing or deleting is allowed but they can be copied. This is to ensure the integrity of the template. New variables can be added to a template and new revisions can be created. These new variables are open for editing, copying or deletion, unless the program version is locked.

TABLE VARIABLE OPTIONS

- Creating a Table Variable
- Editing a Table Variable
- Copying a Table Variable
- Deleting a Table Variable
- Linked Variables
- Working with Table Data
- Creating a Table Variable Revision
- Changing the Active Revision of a Table Variable
- Deleting a Table Variable Revision

Navigating to Table Variables

There are three paths to view or manage variables, either from the Universals tab, Globals tab or from the Program Details tab. The universal tab displays universal variables only. The global tab displays global variables only. You cannot view or manage local variables on the Global tab or the universal tab. On the Program Details tab, you can select to view either global variables or local variables. If you want to view and manage local variables, you must use the Program Details tab. The views and options are the same on either tab.

When you select Table Variables in the Program Details Folder, the current table variables are listed in the Folder Details area.

Table Variables is not an option on the Program Explorer tab.

1. Modules→Program Explorer to open the Program Explorer tab.

2. Select the domain type→domain→project→product→folder→program→program version. You also can access table variables from the globals tab. Domain type→domain→project→product→ globals or the universals tab via domain type→domain→project→universals.

3. Open the program version to get the Program Details tab. Variables are available from the Program Details Folders area on the left side of the tab.

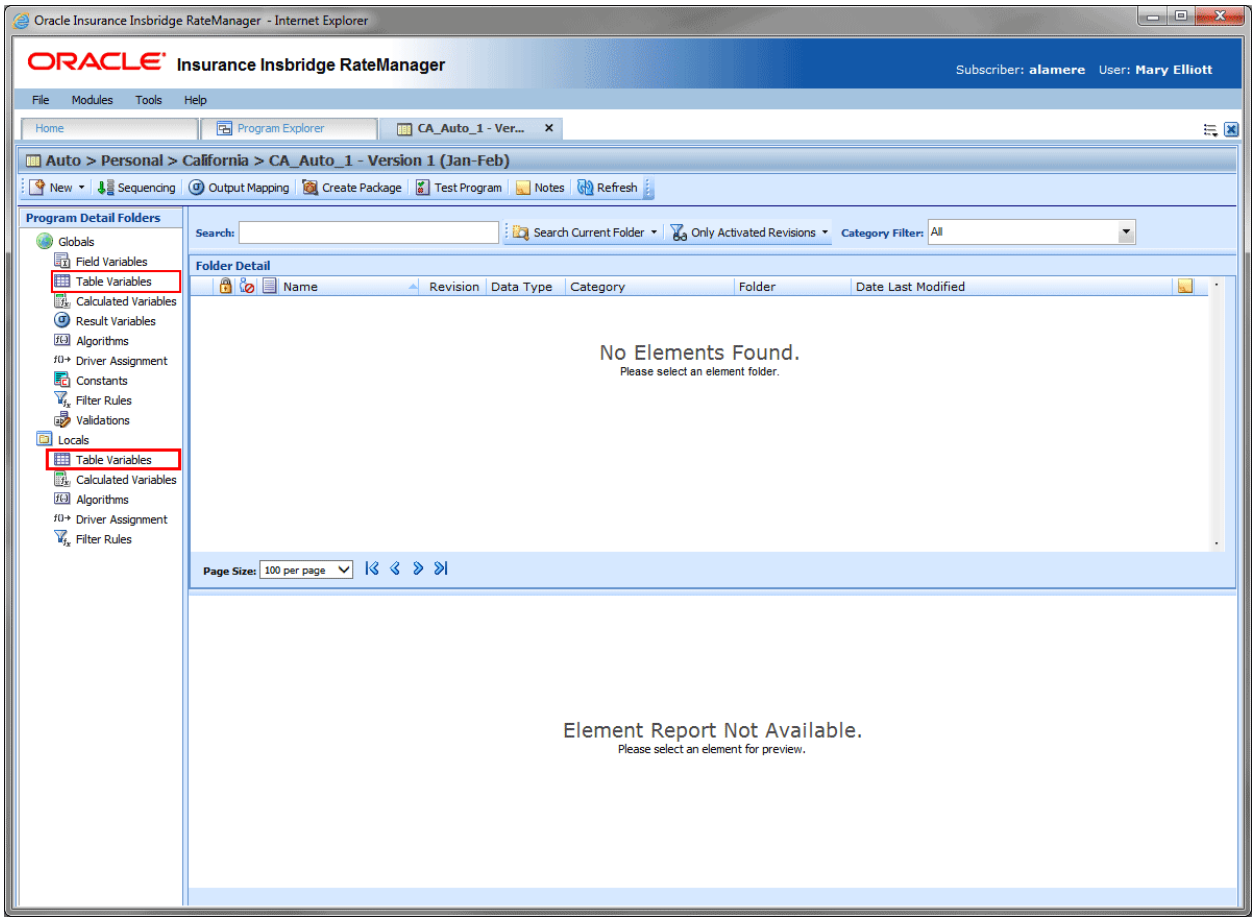


Figure 172 Table Variable Navigation

TABLE VARIABLES FUNCTIONALITY ON PROGRAM DETAILS TAB

On the Program Details tab and the universals tab, available functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any unlocked table variable.

- **Table Variable Functions:** The copy and delete functions can be performed on the Program Details tab using a right click menu. Reports also can be generated using the right click menu on the Program Details tab.
- **Table Variable Revision Functions:** You can view, copy, and delete table variable revisions. Active revisions can be set on the Program Details tab as well.
- **Table Variable Tab:** When you create a new table variable, edit an existing variable, work with table data, or work with revisions, you are placed on the Table Variable tab.

The functions you can perform depend on the status of the table variable. Locked table variables have fewer options. There are icons that can tell you at a glance what the status of the table variable is.

Icon	What it Indicates
	Table Variable. Universal table variables are project specific and can be created on the universal tab only. Global table variables are product specific and can be created at the global level, or from a program version. Local table variables are program specific and can be created at the program version level. The same options, new, edit, copy and delete are available on universal, global or program details tab.
	Data Source. Universal data sources are project specific and can be created on the universal tab only. Global data sources are product specific and can be created at the global level, or from a program version. Local data sources are program specific and can be created at the program version level. The same options, new, edit, copy and delete are available on universal, global or program details tab.
	Universal. Indicates that the table variable was created at the universal level. The table variable is grayed out and can be edited, copied or deleted from the universal tab only.
	Locked Variable. Locked table variables are not open for editing. A lock icon indicates that this table variable is locked by a program and is not open for copying, editing or deleting. If the table variable is locked because the program is locked, you must return to the program and unlock it. This will unlock the table variable and changes can be made.
	Template Lock. If the table variable is locked because it came from a template, you will not be able to unlock the variable. You can create a new table variable if necessary.
	In-Use. Variables that are in use are currently being worked on by another user in the system or variables that you have open in another tab are not open for editing, copying, or deleting. The variable will be available when the user exits the variable.
	Advanced Option. This variable has a wildcard applied to it.
	Active Revision. This variable is the current active revision.

NOTE: *You can search Table Variables by name or by using the Variable Request ID number. Table Variables being searched by Variable Request ID must include the text "ID:" The format is: "ID:(number)".*

Program Details Table Variables Details

The screen details are the same for the universal tab, global tab and program details. For more on screen detail descriptions please see Program Details→Screen Details.

TABLE VARIABLE TAB

The Table Variable tab is available when you create a new table variable, edit an existing variable, work with table data, or work with revisions. You also can be placed directly on the Table Variable tab from the *My Recently Opened Table Variable* snap on your Home tab.

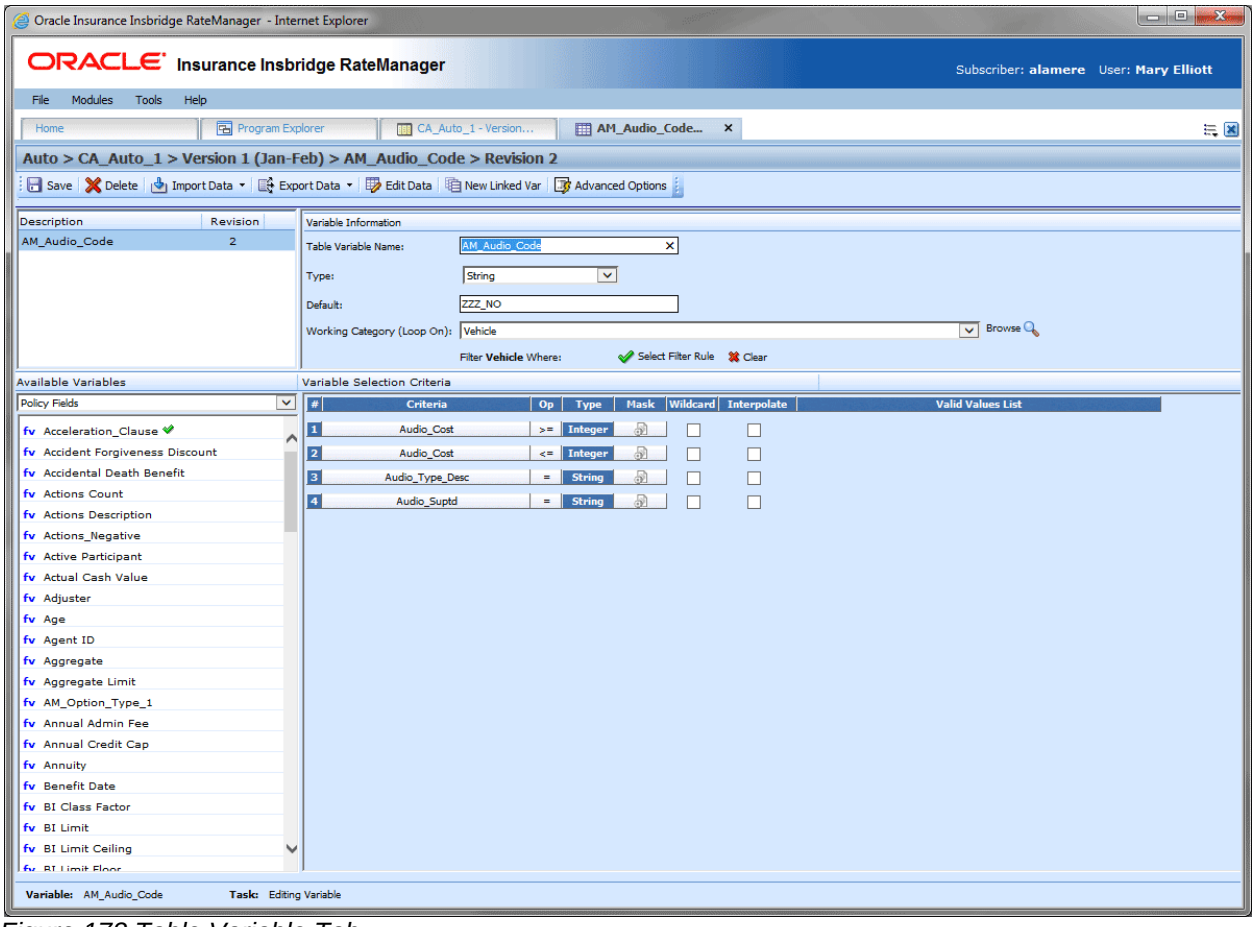
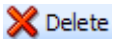
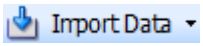
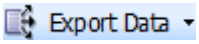
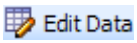
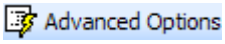


Figure 173 Table Variable Tab

TABLE VARIABLE TAB FUNCTIONALITY

The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform on this tab.

Functions

	Save: Saves the table variable.
	**Delete:** Either removes the selected criteria from the table variable or deletes the selected linked variable from the program. If the variable is still being used in the program, you will receive an error message.
	**Import Data:** Allows the user to import a tab-delimited data file or a Microsoft Excel workbook. See Importing a Table for more information.
	**Export Data:** Allows the user to export the contents of the current table variable or linked variables to a tab-delimited file or a Microsoft Excel workbook. See Exporting Data for more information.
	**Edit Data:** Shows the data table for the current table variable. The data table can be used for verification and quick editing purposes. Data can be changed and rows added or deleted from this tab. See Editing Data in a Table for more information.
	New Linked Var: Creates a new table variable that is based on the same criteria as the currently selected table variable. The new variable will be "linked" with the current table variable and data can be updated simultaneously.
	**Advanced Options:** Changes how RateManager handles fields that are NULLs/Wildcards. By default, RateManager attempts to make an exact match and in the absence of an exact match, RateManager will choose the row that has the most matches. Activate this option if you want RateManager to choose the first possible row, even if a better match exists later in the table. ***NOTE:*** *An advanced option is only available if at least one of the criteria has the wildcard option selected.*

NOTE: *If a change is made to a variable, the change affects all programs that use that variable. If a change only needs to be made to one particular program, create a local variable in the program and have the program reference the local variable instead of the global or universal.*

TABLE VARIABLE TAB SCREEN DETAILS

There are four areas on the Table Variable Tab.

- **Linked Variables Listing:** If this is a linked variable, this area will list all the variables that are linked together. You can select the linked variable to view by clicking on the variable name.

- **Variable Information:** The variable information area contains the name and defines the data type, default value and working category of the table variable.
- **Available Variables:** The list of the available variables and constants.
- **Variable Selection Criteria:** The criteria selection area is where you define what the variable is based upon. For example, if you have a territory code look-up that is based upon ZIP code, then you would select your ZIP code variable to be the criteria.

Table Variables Linked Variables Listing

Linked variables are a group of table variables that are based on the same criteria. For example, if you had two different base rates that were both based off the same table of territory codes, then you could link these two variables.

When you create a linked variable, you do not need to define criteria in each variable. All linked variables use the same criteria.

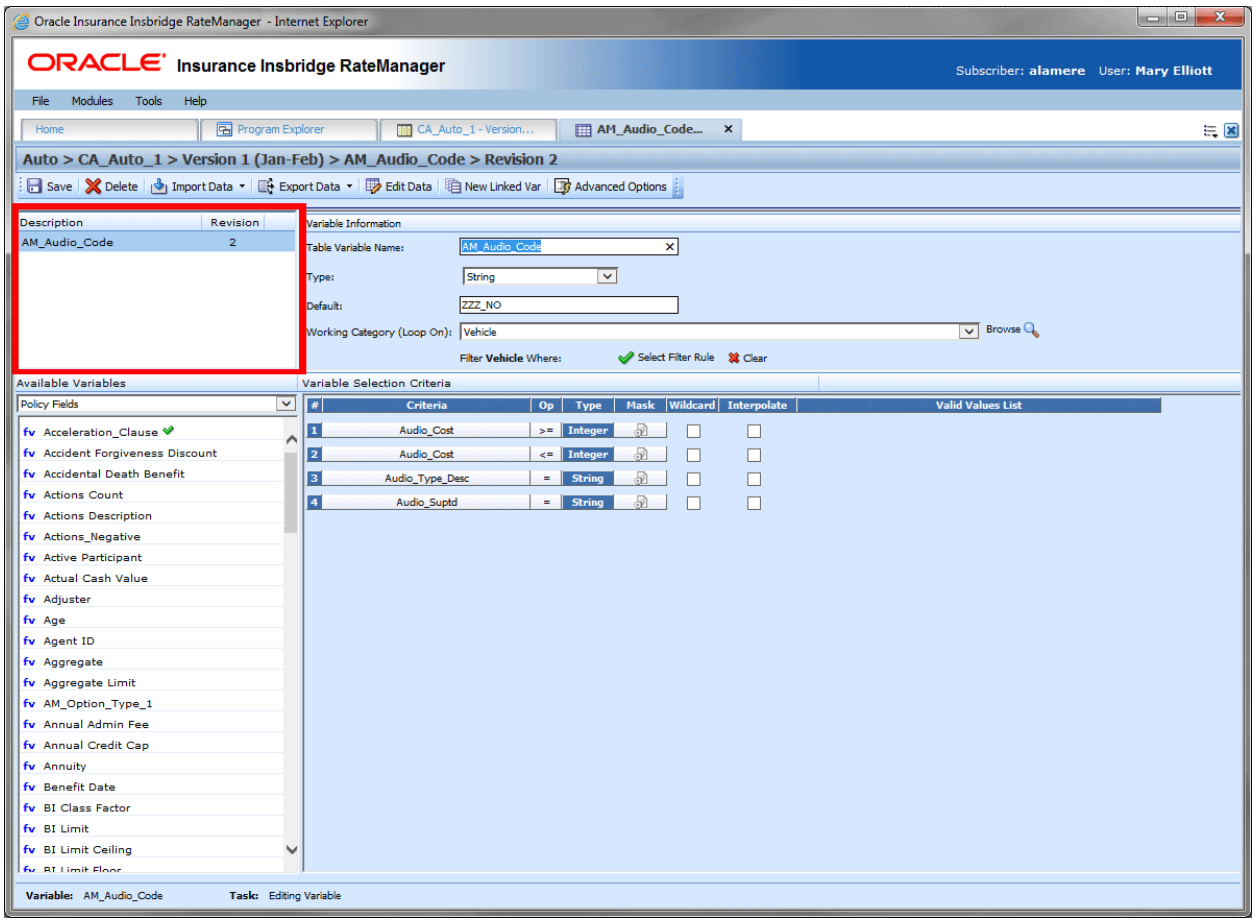


Figure 174 Table Linked Variable Field

Screen Details

Description: Displays the name of the current table variable and all other linked variables.

Revision: Displays the revision number of the current table variable and all other linked variables.

Table Variables Variable Information

The variable information area is where you define the table variable.

Naming standards can be applied to system elements. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the element and for case enforcement to be applied to any newly created element or any existing elements that are being edited. If a naming standard is in place, the pattern is displayed in the element name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented for all elements.

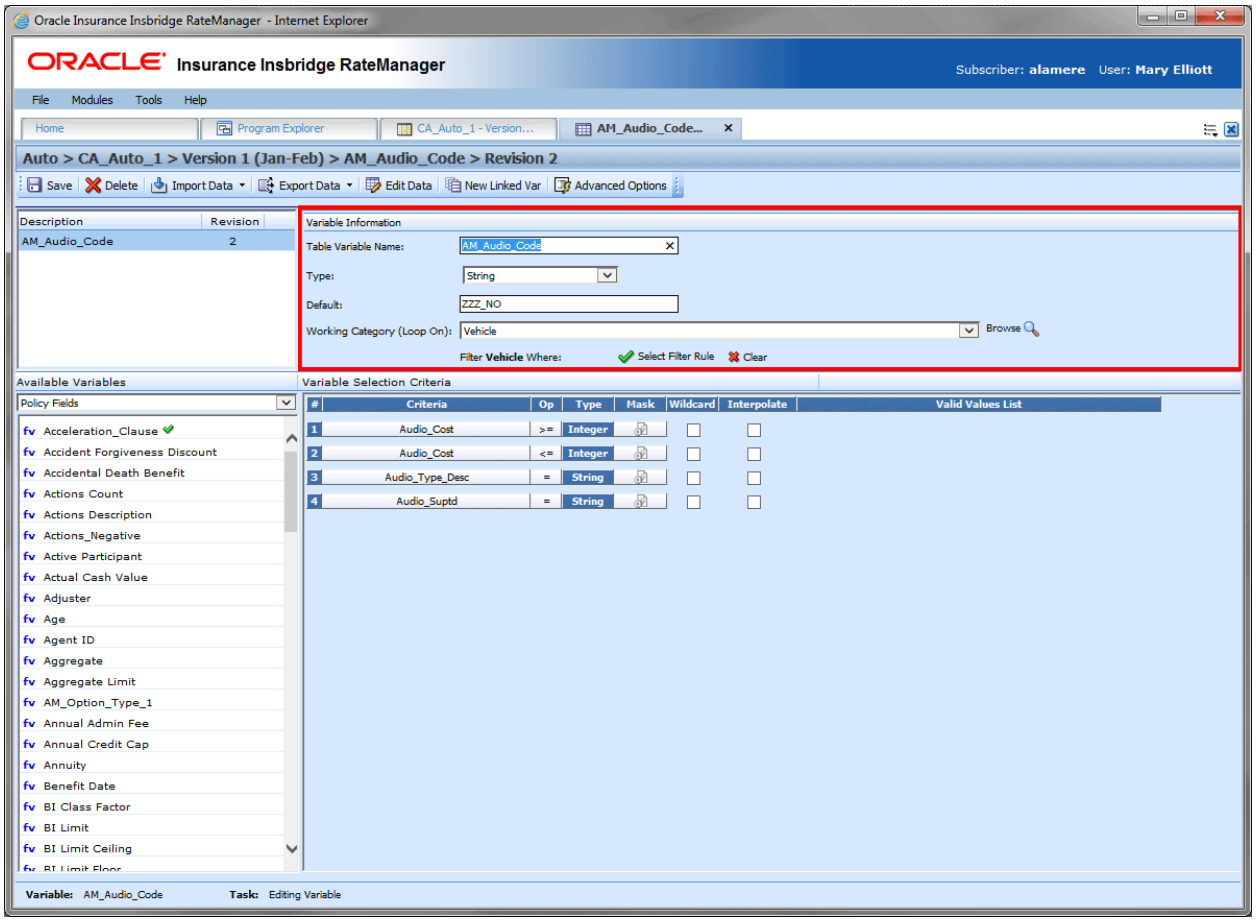


Figure 175 Table Variable Information Field

Screen Details

Table Variable Name: Entry for what you want the variable to be called. Alphanumeric characters are permitted, but no special characters (ex: &, *, ", +, @, etc.) are allowed. In the case of linked variables, each variable has its own name. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

Data source names follow the same rules.

Type: Type of data associated with the variable. Options are:

- **Integer:** Used for whole numbers
- **String:** Used for any alpha numeric characters, including most special characters
- **Decimal:** Used for numbers with decimal points
- **Date**: Used for dates

In the case of linked variables, a different data type may be set for each. If disabled by the system administrator in the configuration file, the data type cannot be changed for any existing variables.

Default: Sets the default value of the variable. A default value is the value assigned to the variable if a match is not found in the variable table. If your table is designed so that there always is a match, then set the default value to either a very low number (0) or a very high number (999). By setting the default value to an extreme number, you can see during testing that the table does not have a match.

You also can set the default value to a valid value. For example, if most of the cities in a state get a territory code of 1, then you could set the default value to 1, and only have entries in the table for the cities that have a territory code other than 1.

For linked variables, a different default may be set for each.

Table Variable Default Value will not allow:

Name	Character
Apostrophe	'

All other characters are allowed in a table variable default value.

Working Category: Sets the category where this variable will be run. For example, if you have an auto test case with three vehicles, any variable set in the vehicle category will be run three times, once for each instance of the vehicle.

You can select the working category from the drop down menu or you can browse for a working category by clicking **Browse**. This brings up the Browse Category window. From here you can navigate to the Category you want. Once you have selected the category, click OK. The category auto fills in the working category section.

The Working Category determines the options displayed in the Available Variables area. For example, if you select Policy as the Working Category, then the menu choices in the Available Variables area are Constants, Policy Fields, Policy Variables and Policy Results. If you select Vehicle, the options now include Vehicle Fields, Vehicle Variable and Vehicle Results along with Constants, Policy Fields, Policy Variables and Policy Results.

Data sources are not used in other variables and working categories are not used. It is recommended to keep data sources in the policy category.

Filter Rule: Filter rules are assigned to table variables, calculated variables, or algorithms to determine if the table variable should be run. Click the green checkmark to add a filter rule. The rule can be removed by clicking the clear option. When a rule has been selected, the rule name is listed.

Data sources do not have the option of filter rules.

Table Variables Available Variables

The Available Variables area contains a listing of available fields, variables, and results sorted by category and select system constants that can be used as criteria. For universal variables, only other universal variables may be used. For global variables, only other global variables may be used. You cannot use any local variables in the creation of a global variable or a universal variable. A globe icon is used to indicate a global variable.

Data sources only have the option to create columns. Other variables cannot be used in a data source.

For Fields, you can add a new field variable without leaving the Edit Table Variable tab. Scroll to the bottom of the available variables list and select <New Field Variable>. A window is displayed where you can enter a new field variable.

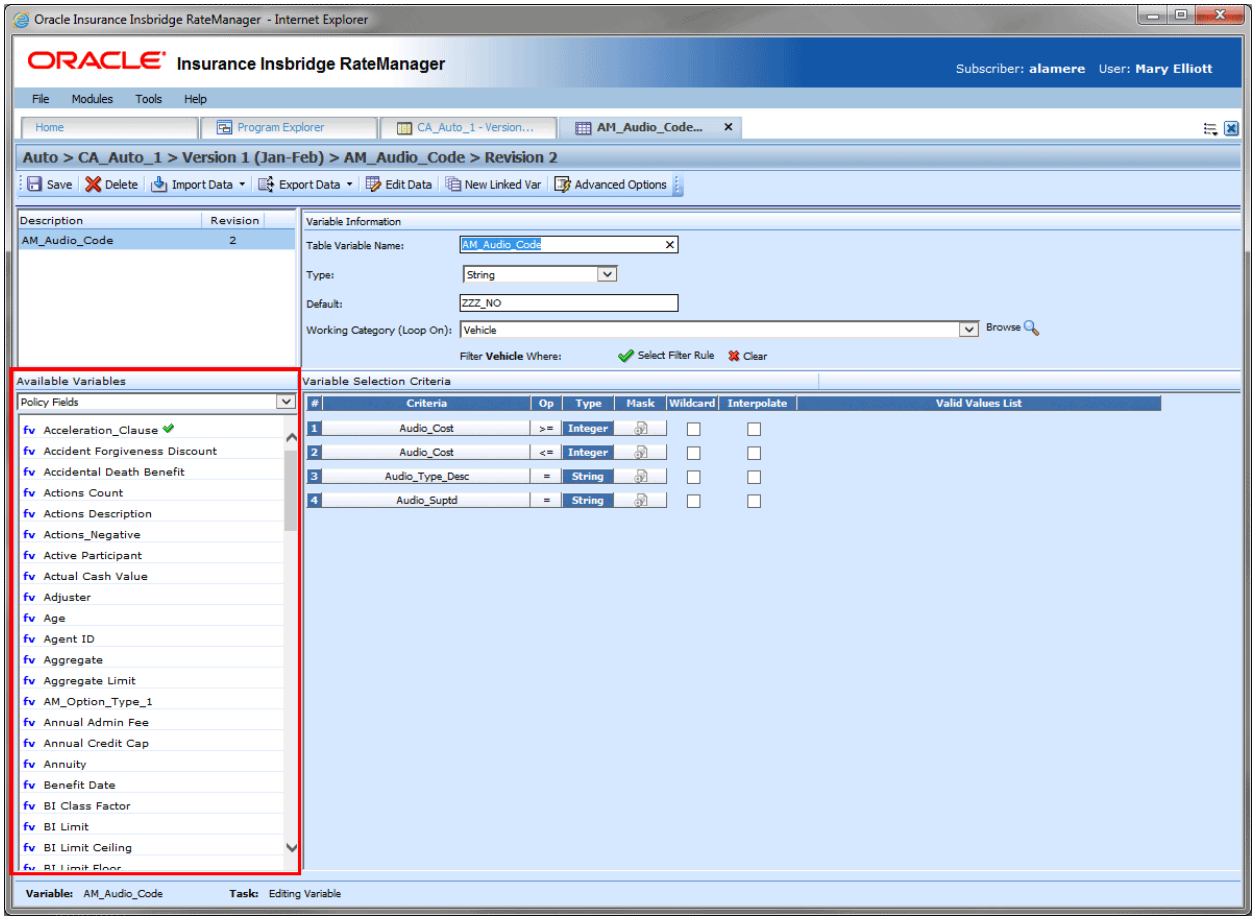


Figure 176 Table Available Variable Field

Screen Details

Menu: The menu is the top most field. Click the down arrow to select the category you want. Policy variables are available for all categories. Driver and Vehicle categories are available in the Driver-Vehicle category. Categories are determined in the Variable Information area.

Selection: The available variables are listed based on the option you select from the menu. The available constants are system level information types, for example current system date or current program version.

Table Variables Variable Selection Criteria

The criteria for the table variable are defined in this section. The Variable Selection Criteria determines whether or not this variable is used. The criteria available are listed in the Available Variables area. You can select fields, variables or results from a selected category or system constant or domain constants.

The order number, criteria, operator, and type are populated with your selection. You cannot change the order number or the type. If you want the criteria listed in a specific order, you must select the criteria in the order you want. All criteria are used in assessing whether or not to use this variable.

NOTE: *A maximum of 50 criteria can be added to a table variable.*

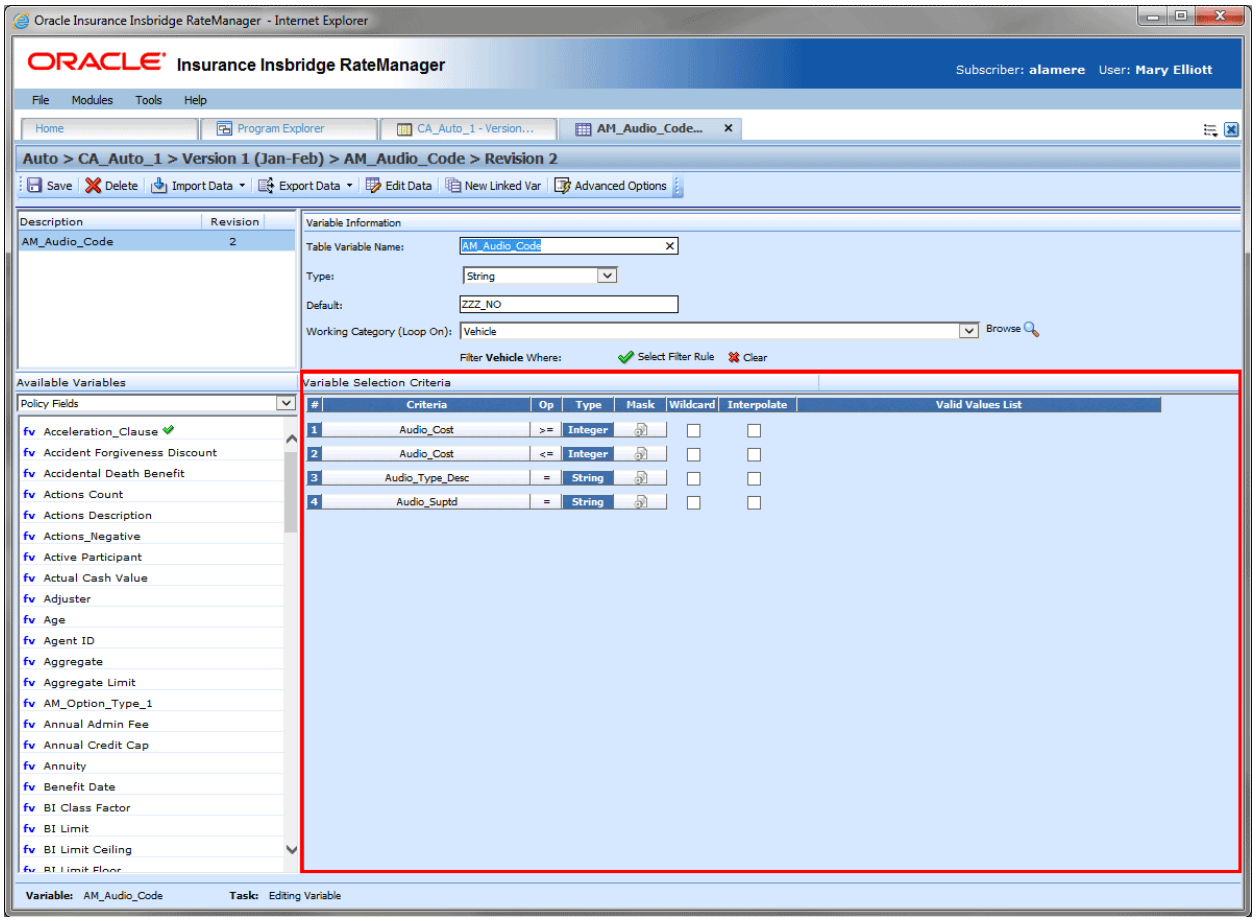


Figure 177 Table Variable Selection Criteria Field

Screen Details

#: Sequential order of criteria selected from **Available Variables**.

Criteria: Displays the name of each variable selected as criteria for this table variable.

Op (operator): Determines how the data in the table is used. If an operator of <= (less than or equal to) is used and the data table has a value of 5, the program will match on any field 5 or less. To change the operator, select the operator icon and choose a new operator. See Comparison Operators for more details. The default is equals (=).

The **operator** can be changed if needed. The operators **equals** and **not equals** are available for all data types, but only numeric criteria will have **greater than**, **less than**, **greater than or equal to** and **less than or equal to** as options. To see the available options, right click and select the operator you want. Arithmetic comparison functions such as less than (<) and greater than (>) cannot be used on string values. Only the operators of equal and not equal are available for string variables.

Type: Shows the data type of the criteria. The type of data in the table variable's table must match that of the criteria or an error will occur.

Mask: Determines how the data being passed into the program should be interpreted by allowing you to hold, remove or replace characters being passed. (Example: If you only wanted to accept the first, second and third characters of a VIN number being passed, this feature adds the ability to hold these characters and disregard the rest.) Click the icon to set the mask. See Masking for more information.

Wildcard: A check in this field tells the system that one or more rows in the table for this table variable will accept any value passed for these criteria. See Wildcard for more information.

Interpolate: A check in this field instructs RateManager to use interpolation to determine the correct value for the table variable. See Interpolation for more information.

Group: *For data sources only.* Grouping indicates that the qualifier fields are related and may be entered in any order. Grouping the qualifiers together reduces redundant table rows and makes tables easier to manage.

NOTE: *When creating templates, do not use a table variable as criteria for itself. This may result in an error.*

Right Click Menu Options:

The criteria and Op fields have right click menus. Right click the field you want to view to see the options.

Criteria have three options:

- **View Report:** You can view the table variable report.
- **View Where Used:** Displays where this variable is used.
- **Delete**: Deletes the variable.

Op has six options:

- **Equals** =
- **Not equals** <>
- **Less than** <
- **Less than or Equals to** <=
- **Greater than** >
- **Greater than or equals to** >=

Data sources list column criteria. There are only two right click menu options, edit and delete.

VARIABLE SELECTION CRITERIA

The variable selection criteria can have one or more options applied to it.

Table Variables Masking

Masking determines how the data being passed into the program should be interpreted. For example, if only the first five digits of the VIN number are needed, you can use the masking function to read only the first five characters of the value.

	NO Mask: No mask has been set for these criteria.
	Mask: A mask has been set for these criteria.

To Set a Mask:

1. Click the mask icon . This brings up the Mask Value screen.
2. Enter in the Mask Value.

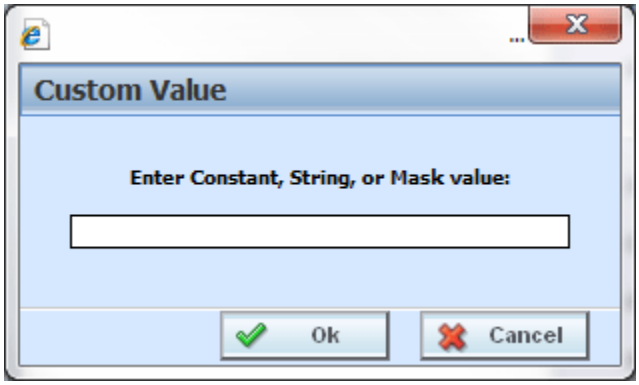


Figure 178 Mask Value Dialogue Box

3. Click **OK** to save the entry. You will be returned to the Edit Variable tab. The mask icon now reflects that a mask has been set with a change in color .

Available Mask Options:

Function	Definition	Keystroke(s)
Hold	Holds a character, i.e. keeps the character	~
Remove	Removes a character	|
Replace	Replaces a character with another character	Any character
Insert	Adds a character	^ + Any character

Examples:

Function	Input	Mask	Result
Hold	GA263SX4597	~~~~~	GA263
Remove	VEH01	|||~~	01
Replace	Insbridge	~~~urance	Insurance
Insert	Cat	~^h~^r~	Chart

Table Variables Wildcard

Wildcard is a function used when certain data in your table is only relevant in specific cases. For example, if Territory Code is determined by GaragingZip and, in some cases, ResidenceType, a value or an empty field could be used in the ResidenceType column.

The empty field signifies that anything being passed in this field is an acceptable match, as long as the wildcard option has been selected for that field. A specific value will only assign that territory if an exact match is made.

Wildcard Table

Territory Code	GaragingZip	ResidenceType
1	65478	2
2	65478	
3	65435	
4	65471	

If the Wildcard table above is used, only the exact input values of GaragingZip 65478, and ResidenceType 2 would produce a Territory Code of 1. GaragingZip of 65478 with any other ResidenceType will produce a Territory Code of 2.

WARNING: ***The wildcard function MUST be checked on the table variable tab if empty fields are used within the table. Otherwise, an incorrect match and error will occur.***

NOTE: *When importing data for a table variable, use an asterisk to signify a field that is wild carded. Asterisks help insure the data gets imported properly into RateManager.*

Table Variables Interpolation

To interpolate means to estimate a value between two known values. For example, if you know that the Key Factor for a DwellingLimit of 10,000 is 2.00 and the Key Factor for a DwellingLimit of 20,000 is 3.00, you can use interpolation to figure out that the Key Factor for a DwellingLimit of 15,000 is 2.50. But what is the Key Factor for a DwellingLimit of 13,500? To find the Key Factor, input your table with the known values and select the checkmark under **Interpolate** next to the criterion you want to interpolate.

Here are the steps used to calculate the Key Factor for a DwellingLimit of 13,500 from the example.

Steps	Action
1. Calculate the difference between the known Key Factor's to get a factor per 10,000 (since the difference between 10,000 and 20,000 is 10,000)	3.00 - 2.00 = 1.00
2. Divide the factor per 10,000 by 100 to get a factor per 100	1.00 / 100 = 0.01
3. Determine the number of 100's between 13,500 and 10,000	13,500 - 10,000 = 3,500 / 100 = 35
4. Multiply the factor per 100 by the number of 100's	0.01 x 35 = 0.35
5. Add the increased factor to the Key Factor for 10,000	0.35 + 2.00 = 2.35
The Key Factor for a DwellingLimit of 13,500 is	**2.35**

Usage

- Interpolations should be applied to non-linked variables only. Linked variables may not function as expected.
- Interpolate should not be used in a table with a wildcard. Conflicts may arise when combining a wildcard with an interpolated value in the same table.
- Only one variable selection criteria element should have interpolation applied.
- Interpolation only works with integers and decimals. The criteria have to be a number, and the table itself has to be a number. You cannot interpolate on a string or date.

CREATING A TABLE VARIABLE

When you create a new table variable, it gets created in all versions of a program. Table variables can be added to both standard and templates programs. There is no limit to the number of variables that can be created for a program.

1. To create a universal variable, select the universal icon under the project where you want to create a variable on Program Explorer. Click New, select Table and then select Table Variable. To create a global or local table variable, enter the program details tab from the program where you want to create a variable. In the top bar menu, select New, select either Globals or Locals, select Table and then select Table Variable.

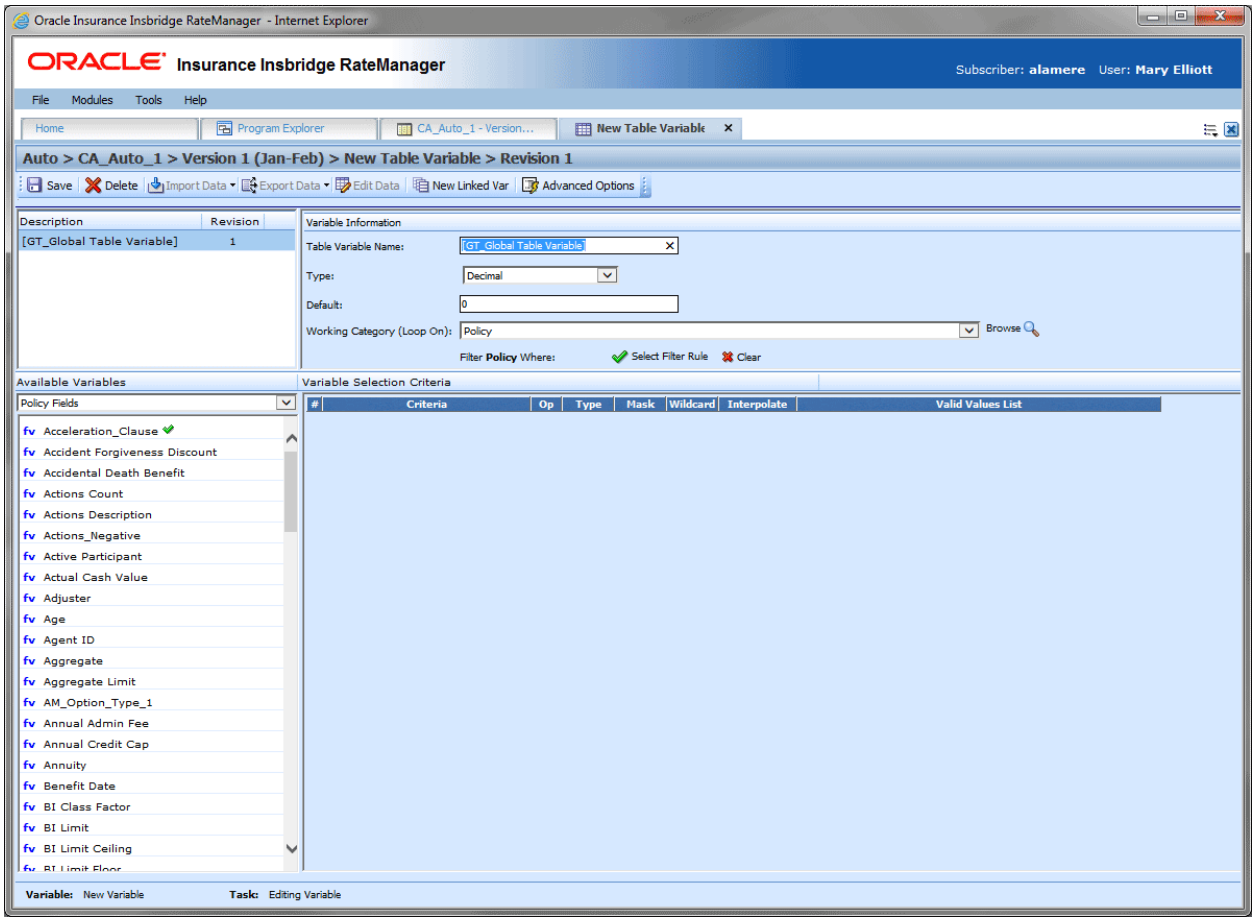


Figure 179 Creating a Table Variable

2. Enter in a **name**. Alphanumeric characters are permitted, but no special characters (ex: &, *, ", +, @, etc.) are allowed. In the case of linked variables, each variable has a unique name. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Select the **type** of data associated with the variable. Options are:

   - **Integer**
   - **String**
   - **Decimal**
   - **Date**

In the case of linked variables, a different data type may be set for each. If disabled by the system administrator in the configuration file, the data type cannot be changed for any existing variables.

4. Set the default value of the variable. The Default Value must match the data type selected and the field is limited to 50 characters.

5. Select the working category.

6. In the Variable Selection Criteria area, select an available variable from the drop down menu and double-click it. This adds it to the variable selection criteria area. You can enter up to 50 criteria.

7. Check the filter rule option to add a filter rule. See Adding Filter Rules for more information.

8. If you want to apply a mask, click the mask icon. See Masking for more information.

9. Check the wildcard box for any variable where you want the system to accept any passed criteria. See Wildcard for more information.

10. Check the interpolate box if you want to use interpolation to determine the correct value for the table variable. See Interpolation for more information.

11. Click Save to save your work.

Adding a Variable or a Custom Value

A new field variable can be created while entering or editing a table variable without leaving the Edit tab.

- To add a field **Variable**: See Adding a New Field from Another Area

LINKED VARIABLES

Linked variables are a group of table variables based on the same criteria. For example, if two different base rates were both based off the same table of territory codes, these two variables could be linked. RateManager supports a maximum of 100 variables associated with one single table. This means that a linked variable can have itself and 99 other variables.

Variables can be linked at the time you create them or you can enter new linked variables to an existing variable at a later time on the Edit Table Variable tab. Existing variables cannot be linked together. For example, if you created Table Variable A and then created a separate Table Variable B, you will not be able to link them together.

Creating a Linked Variable:

1. Select the table variable you want to add a linked variable to. Right click and select **Edit**. This opens the Edit Table Variable tab. If you are creating a new table variable, you can add a linked variable at any time in the create process.

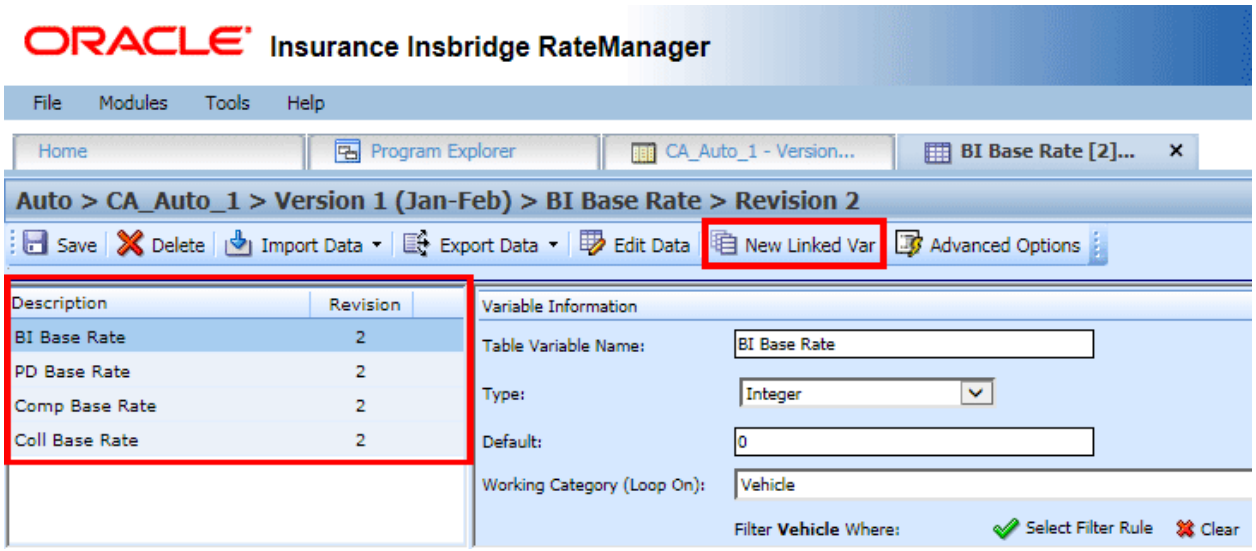


Figure 180 Creating a Linked Table Variable

2. Click **New Linked Var** from the top bar menu. The Variable Information area at the top right side portion of the tab will refresh. Another variable will be added in the Linked Variable Listing at the top left side portion of the tab.

3. Enter in a name, select the data type, a default, and the working category.

4. Table variables that are linked are listed in the Linked Variables Listing area on the left. You can return to any of the variables listed by clicking it.

5. Click **Save** to save your work.

EDITING A TABLE VARIABLE

The **Edit Variable** tab changes information about the variable such as the name, data type, default value, working category and variables used as selection criteria. It also is where you can edit the data table, import a table, export the table and create new linked variables.

Template generated variables that are using revision 1 are not open for editing. If you want to edit a template generated table variable, you must create a new revision. The new revision is open for editing except for the name. The name cannot be changed.

1. On the Program Details tab, select the **Table Variable** you want to edit.
2. Double click or right click and select **Edit**. The Table Variable tab opens with the details of the table variable.
3. Make any changes to the variable information or table.
4. Click **Save** to save your work.

WARNING Category Updates to Revisions

Category updates should not be done to a revision. All revisions of a variable should be at the same category level. Changing the category for only some of the revisions of a variable is not supported.

TABLE VARIABLE COPY OPTIONS

Table variables may be copied in any program.

- **Local table variables** can be copied to the same program or to another program at the local level within the same product.
- **Global table variables** can be copied:
  - At the global level and are available to every program in the product
  - To the local program level
  - To another program within the same product at the local level
- **Universal table variables** can be copied:
  - At the universal level and are available to every program in the project
  - To the local level of any program within the project

The copy options depend on whether the table variable to be copied is at the universal level, global level or the local level.

Copy Options

When a table variable is copied at the same level, there are two copy options; Copy Definition and All Data or Copy Definition only. When a table variable is copied to a local level there are four copy options:

- **Copy Definition & All Data:** The table variable and all associated table data are copied to the new variable.
- **Copy Definition Only:** Only the variable definition and the structure of the table variable (including the data type, default value, working category and criteria definitions) are copied, but not the data.
- **Other Program - Copy Definition & All Data:** The table variable and all associated table data are copied to the new variable.
- **Other Program - Copy Definition Only:** Only the variable definition and the structure of the table variable (including the data type, default value, working category and criteria definitions) are copied, but not the data.

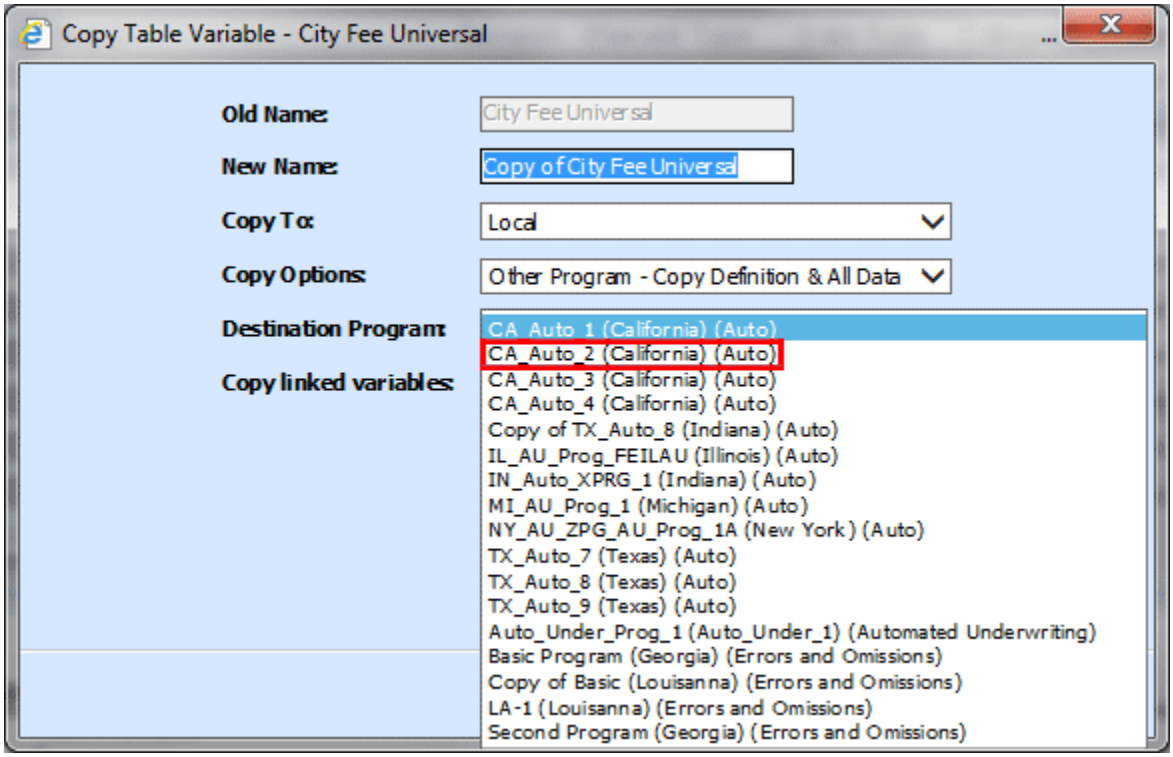


Figure 181 Universal Table Variable Copied to a Local Level

When copying a universal table variable to a local level:

- Universal copies to a local level are project wide:
  - Information has been added to help you select the Destination Program where you want to copy the element to. The Destination Program name is listed first, the folder name is in parenthesis and the product name is last, also in parenthesis.

  - In Figure 181 Universal Table Variable Copied to a Local Level, the highlighted program is CA Auto 2, in the California folder, under the Auto product. The Destination Program name is listed first, the folder name is in parenthesis and the product name is last, also in parenthesis.

- Duplicate naming is not allowed. The New Name must be different from the Old Name.
- Product level naming standard are not enforced. If the product has a naming standard, copies made from the universal level are not required to have the naming standard applied when copying. If the variable is edited any time after the initial copy, the naming standard is enforced.
- Universal permissions are required to make copies at the universal level.

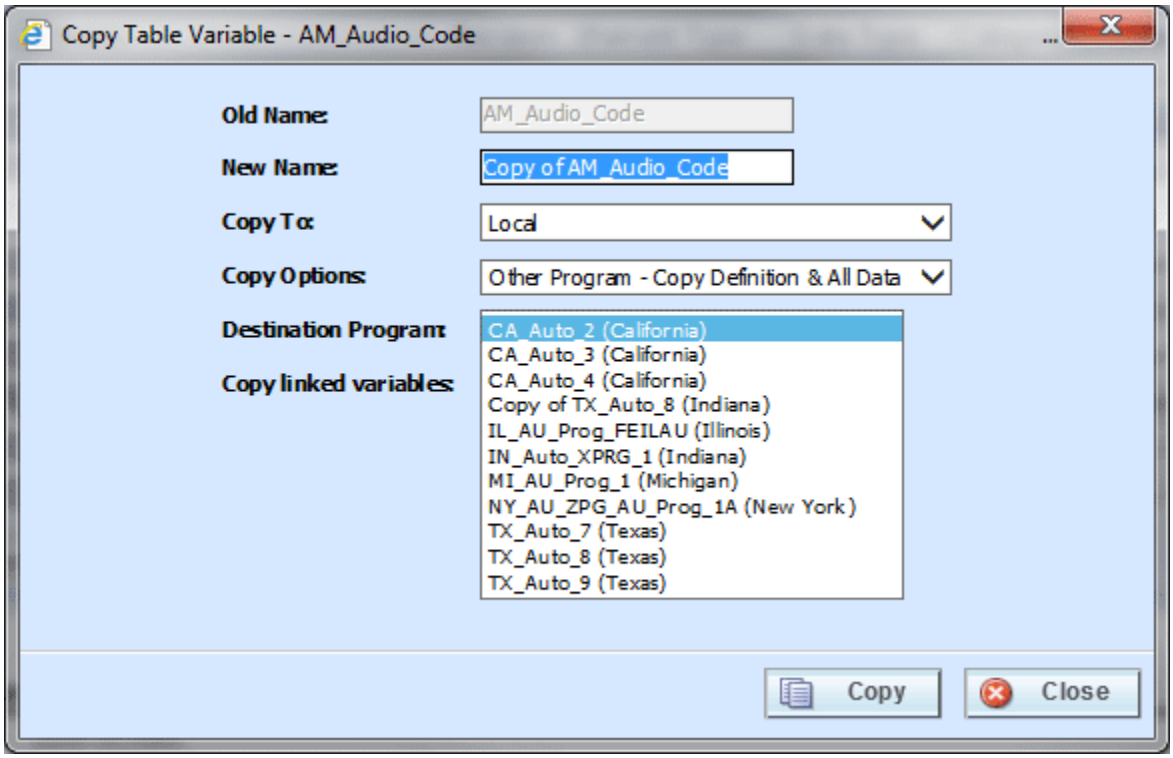


Figure 182 Copying Table Variable at the Global Level

When copying a global table variable to a local level:

- Global copies are done at the product level. The Destination Program name is listed first and the folder name is in parenthesis.
- The New Name must be different from the Old Name.

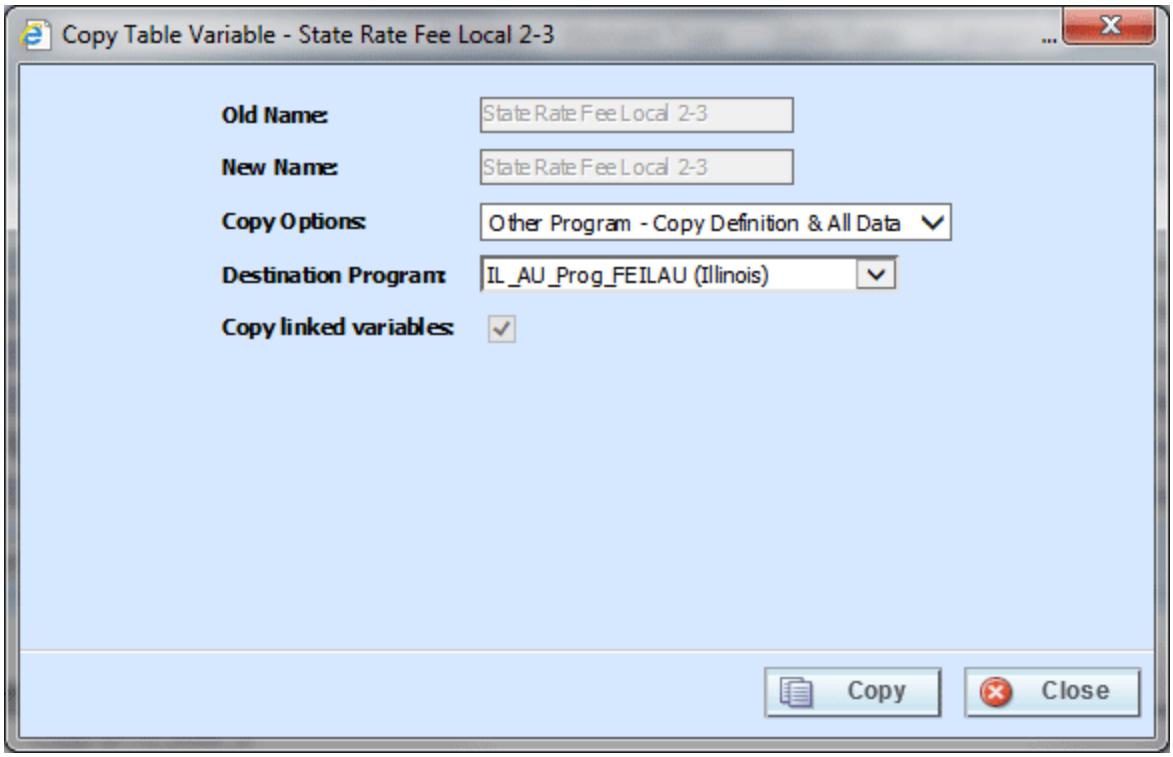


Figure 183 Copying Table Variable at the Local Level

Local copies can be done to the local level of any program in the product. When copying a local table variable to a local level in another program:

- Local copies are done at the product level. The Destination Program name is listed first and the folder is in parenthesis.
- Local to local copies check for dependencies when the copy is made:
  - When performing a local to local copy, dependencies such as the usage of other local elements are checked. If there are local elements used in the element being copied that are not in the Destination Program, then those elements are copied also.
  - For example, the Option AB Fee algorithm, a local variable, uses table variables Option A and Option B, also local variables. When Option AB Fee is copied from program Maine to program Kansas, the algorithm is copied and so is the table variables that are being used.
- Local to local copies do not require a name change:
  - When copying to the local level of a different program, the table variable name cannot be changed. The New Name is grayed out. This creates an identical variable in the other program and reduces the need for editing. If a change is made to the original table and then the table variable is copied for a second time, the copy is created as a new revision in the Destination Program.
  - When copying at the local level in the same program, the name must be changed.
- Copies made in the current program must have a new name.

Copying Linked Variables

All linked variables associated with the table variable may be copied by checking the **Copy linked variables** check box on the copy table variable window. All linked variables are copied with the name COPY OF… You cannot change the copied variable name on the copy table variable screen. You can rename the variables using the edit option.

If you do not choose to copy all linked variables, the system only copies the linked variable that is being used in the variable you are copying. All other associated variables are not copied.

COPYING A TABLE VARIABLE

1. Select the variable you want to copy and right click. Select **Copy** and a separate window is displayed.

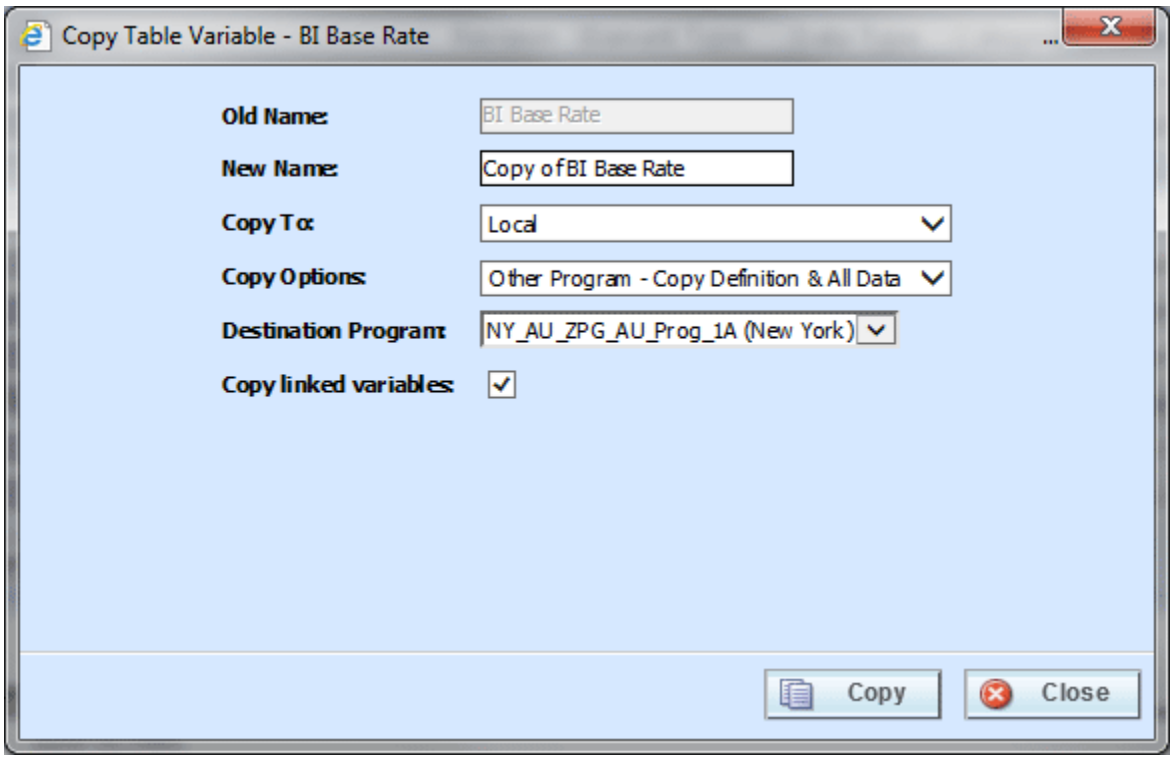


Figure 184 Copying a Global Table Variable

2. Select the **Copy To** level. It is recommended that you change the copy options first. Changing some options may reset the name field.

3. Select the **Copy Options**. The options presented depend on the Copy To level.

4. If you are copying to the local level or to another program, select the **Destination Program** from the drop down menu. Current Program options do not have destination Program options.

5. If there are linked variables and you want to copy them all, check the **Copy Linked Variables** box. If this option is grayed out, no linked variables are available to copy.

6. Enter in a **New Name**. The name can be entered first. It is recommended that you change the copy options first. Changing options may reset the name field.

NOTE: *When copying from the local level to the local level of a different program, the table variable name cannot be changed. The New Name is grayed out. This creates an identical variable in the other program. Linked variables are automatically checked.*

7. Click **Copy** to copy the variable.

8. If the copy was successful, you will be returned to the previous tab. Your variables have been copied. The last updated date is the date the copy was made.

DELETING A TABLE VARIABLE

When a variable is no longer needed, it can either be left in RateManager or deleted. When you delete a variable, it gets deleted from all versions of the program. Also, all revisions of that variable get deleted. If you are uncertain as to whether the variable will be used again, simply leave it in RateManager. This saves time if the variable is needed and does not slow down rating.

1. Select the variable you want to delete. Right click it and select **Delete**.

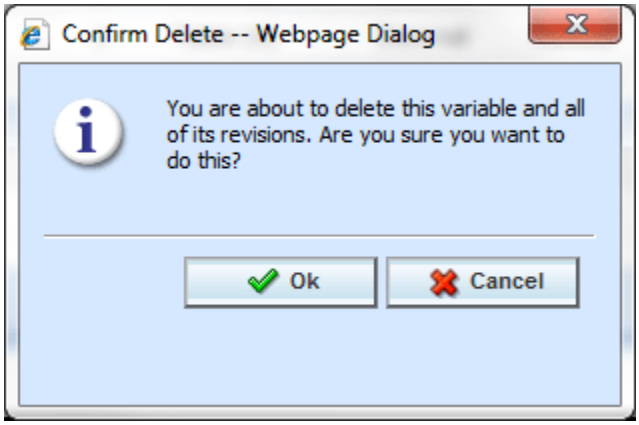


Figure 185 Deleting a Variable

2. You will be asked to confirm the deletion of the variable. Click **Ok** to delete the variable. Click Cancel to return to the tab without effecting a change.

If you try to delete a variable that is still being used by an algorithm, a result group or another variable, you will receive an error. Variables that are locked or that are template generated cannot be deleted.

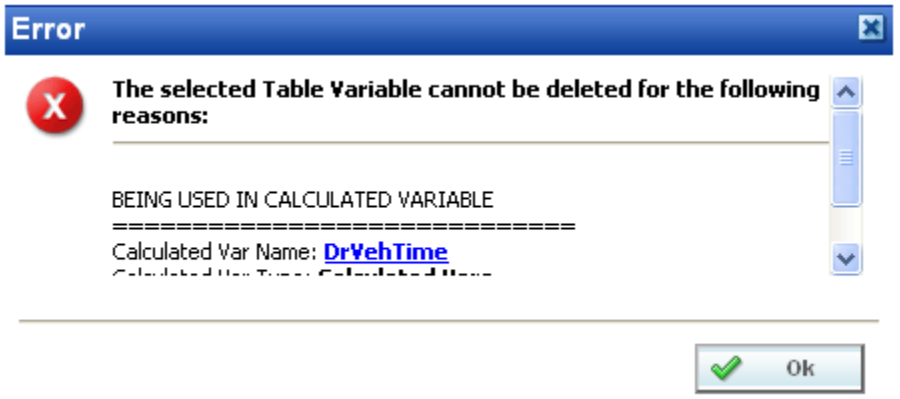


Figure 186 Error when Deleting a Variable

If you delete a linked variable, only the variable you have selected is deleted. The other variables that are linked to it are not removed.

WORKING WITH TABLE DATA

Data can be imported or directly entered in to an unlocked table variable. Data can be exported out of a table variable at any time. Data is managed on the Table Variable edit tab. Data can be imported or exported from either the Table Variable edit tab or by right clicking a table variable on the Program Details tab and selecting Import or Export Data.

There are three buttons on the Table Variable tab to help you with managing data:

- **Import Data**: Used to import data into a table
- **Export Data**: Used to export data out of a table
- **Edit Data**: Used to edit existing data or to enter new data

IMPORT DATA

There are three locations where a table variable can be imported:

- The edit tab of any selected table variable has an Import Data option in the top bar menu that contains a drop down where you can select to import data from a tab delimited text file or from an MS Excel workbook. This option is open to all users with product permissions and allows to you import the exact file you want.

- The program details tab of any selected program version can display the global or local table variables for the program. All table variables have a right click menu with the Import Data option. From there, you can select to import data either from a tab delimited text file or from an MS Excel workbook. This option is open to all users with product permissions.

- The universal tab of any project can display the table variables for the project. All table variables have a right click menu with the Import Data option. From there, you can select to import data either from a tab delimited text file or from an MS Excel workbook. This option is open to all users with Universal permissions. If you do not have universal permissions, you may receive an error.

In order to import data into RateManager, you need an import file. An import file holds the data for a table variable. It is important that the structure of the import table match the structure of the table variable. A few rules should be followed when creating an import file:

- Each result or look-up in the file must be in its own row.

- The first row in the file must contain column headings.

- Each column heading must be unique.

- The number of columns must match the number of table variables and criteria. If you have three linked variables and two criteria, the import file must have five columns.

Two types of tiles can be accepted:

- Tab delimited (.txt)

- MS Excel (.xlsx)

If you want to import data on the Program Details or Universal tab, the active revision is displayed on the listing and this is the revision where the data will be imported in to. If you want to import data in to a table variable revision that is not the active revision, you need to show/hide revisions and select the revision where you want to import data.

Data Preparation

Before importing, look at your table and decide what data you want to import and what type of data it is. You also must decide how you want to use the information in the table. Each column in your table can be brought in as variable selection criteria or as a linked variable.

If the data is based on the same selection criteria, you may want to create linked variables. If the data in your table determines the value to be used, you may want to use fields or variables as selection criteria.

For example, if you have two different rates (Rates A and B) that are both based off the same territory code (TerCode), you could create a table variable (RateA), then create a linked variable (RateB) and use selection criteria field (TerCode). Or you could create a single table variable (RateA) the uses selection criteria field (TerCode) and another separate table variable (RateB) that also uses criteria field (TerCode). In the case where you create two separate table variables, you would have to import your file twice; once to field the data for RateA and a second time to field the data for RateB.

You also may have two territory codes (TerCodeA and TerCodeB) that are used to select a base rate (RateA), you could create a table variable (RateA) with two selection criteria fields (TerCodeA and TerCodeB).

Each column in your table that you want to bring in must be assigned to table variable, linked variable or variable selection criteria. If you do not have a table variable, linked variable or a variable selection criteria, the data will not be brought into the table.

Restricted Use of Special Characters

A table variable file name is the actual name of the file that will be imported into RateManager or exported out of RateManager. For example, 'BI_Rates4-2011.txt'. Table Variable File Names cannot contain:

Name	Character
Ampersand	&
Apostrophe	'
Asterisk	*
Backslash	\
Colon	:
Greater-Than Sign	>
Less-Than Sign	<
Question Mark	?
Quotation Mark	"
Slash	/
Vertical Bar	|

- If you attempt to import a file with a restricted character, you will receive an error message.

- All other characters are allowed in a table variable file names.

Table variable data files are the contents inside the file to be imported into a table variable or exported out of RateManager. Table Variable Data Files will be rejected if restricted characters are contained within the file.

Restricted Characters are:

Name	Character
Apostrophe	'
Ampersand	&
Apostrophe	'
Greater-Than Sign	>
Less-Than Sign	<
Plus Sign	+
Pound	#
Quotation Mark	"
Tilde	~
Bar + H	|H
Bar + h	|h
Bar + L	|L
Bar + l	|l

All other characters are allowed in a table variable data file with some considerations:

- Asterisk (*) must be inside a string (p**l). A single asterisk is considered a wildcard and will be removed. The field is displayed as blank.

- Quotation marks may be placed around commas and quotes. This depends upon the source where your data was created. You can preview your data before importing to visually inspect it. RateManager does not remove quotation marks prior to importing.

- Ticks (`) may not always import properly. It is recommended that you avoid using this character if possible or only use this characters inside a string (an`s).

Data can be imported into any unlocked table variable at any time.

Automatic Mapping of Columns

In the import file, the columns of data do not have to be in any particular order. If you want the columns to automatically match up, place table variables columns first followed by criteria. This allows for the automatic mapping of table criteria and variables to the columns of data. This can save time when importing a large amount of data.

For best results when auto mapping columns, column header names, criteria and table variable names must match exactly, including capitalization.

If the data is not in any particular order or if auto mapping is not selected, you can match the columns in the input file with the variables and criteria in RateManager manually.

Mappings can be changed after the file is presented. This may be necessary if the import file headers differ or are in a different order from table variables or criteria.

Working with Tab Delimited Files

Tab delimited text files are delimited (separated) by a tab character. Tab delimited files can be created using a spreadsheet such as Microsoft Excel or a text editor such as Notepad.

- The file must be saved in tab delimited format using a .txt extension.

NOTE: *Uniform text file format may be required when using special characters. When exporting files created in Microsoft Excel, it is recommended that the file be exported as a Unicode text file.*

Tab delimited files are imported one at a time.

Importing a Tab Delimited File in to a Table Variable

Only the option to automatically map column headers is available when importing a tab delimited file. In this example, a table variable with four selection criteria will be created.

1. Navigate to the **Edit Variable** tab for the table variable from where you want to import a table. Make sure you have any linked variables and selection criteria already entered. **Save** your entry prior to importing.

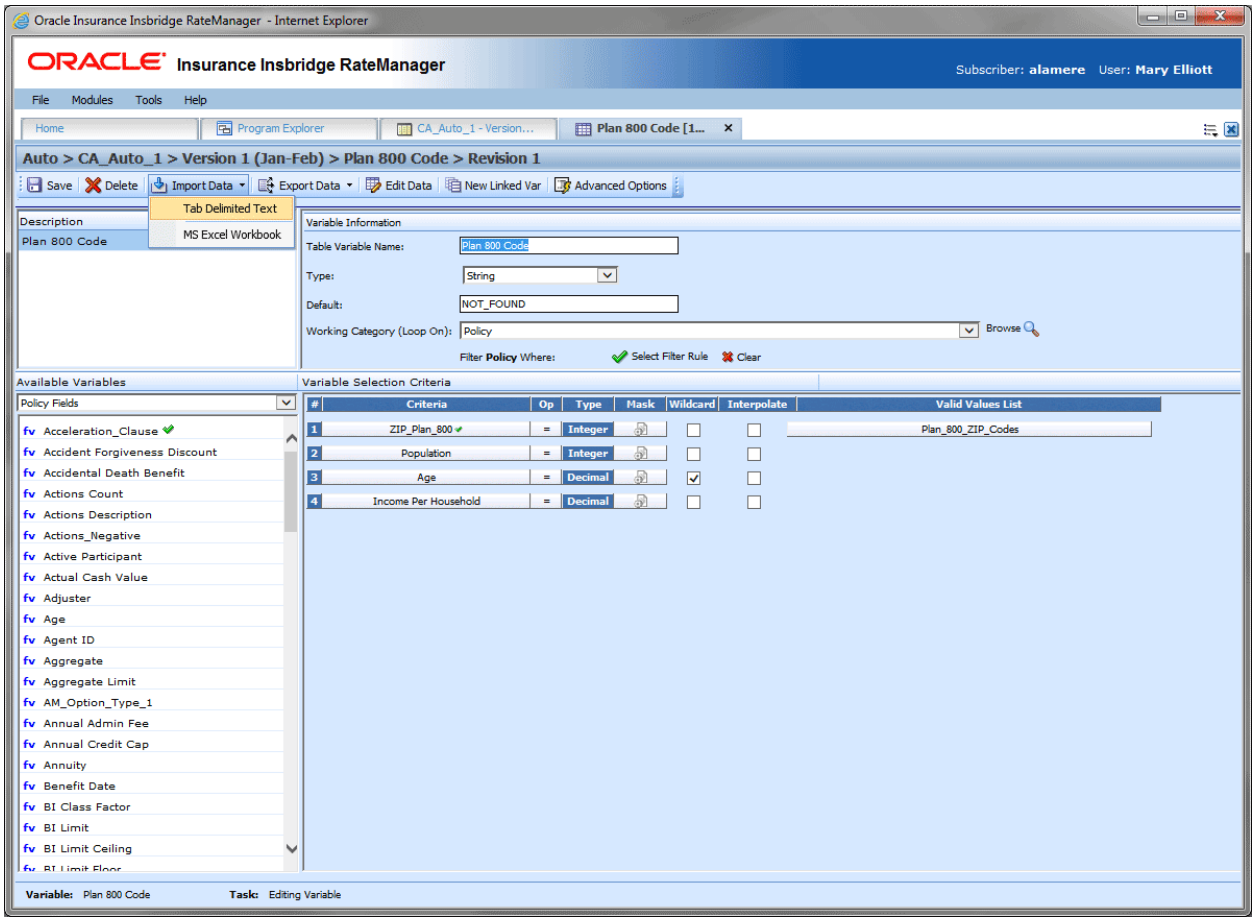


Figure 187 Selecting Data File

2. Click **Import Data** and select Tab Delimited Text.

3. Click **BROWSE…** to locate your file. The automap feature is checked by default. If you do not want the fields to automatically map, uncheck the option. Columns that have been auto mapped can be changed.

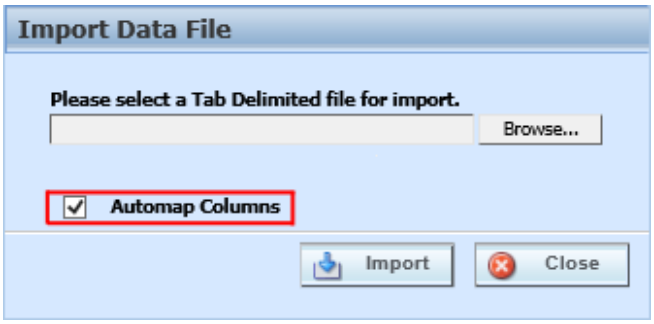


Figure 188 Import Table Data

4. Once the data file has been selected, click Open to begin the import process. The location path of the file is entered in the selection field. Click **Import** to upload the file.

5. You will be asked if you would like to preview your data.

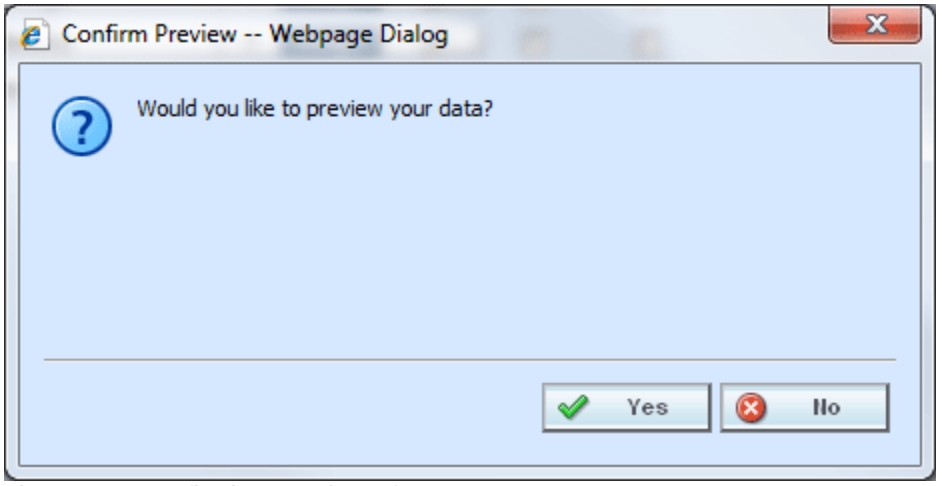


Figure 189 Confirming Preview of Data

6. Clicking **Yes** matches the column headings to the variable(s) and variable criteria automatically and displays the import file data. This allows you to verify the matches are correct.

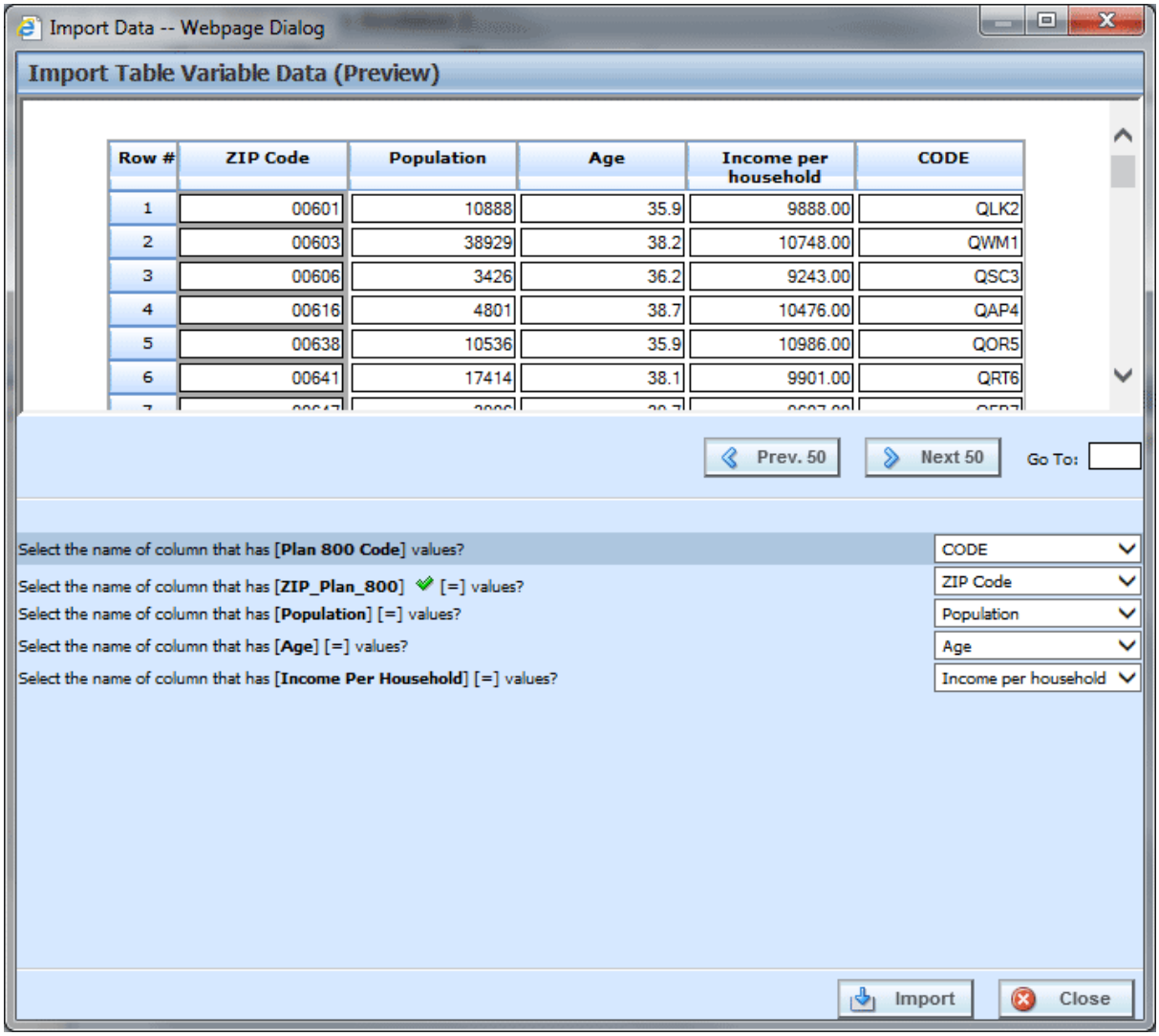


Figure 190 Previewing Data File with Automatically Matched Columns

The import window is split into two parts. The top portion displays the import file data as you created it. The bottom half lists the criteria and variable(s) you created within RateManager. If the column headers, the criteria and table variables names match exactly, the columns will be matched automatically. If there is a difference in the name or position, you will have to match up each element with the correct column in the import file manually.

NOTE: *For best results when auto mapping, column header names, criteria and table variable names must match exactly, including capitalization.*

If the table contains multiple rows, you can use **Next 50** and **Previous 50** to navigate through the table. You also can go directly to a row by entering the row number in the **Go To box** and clicking anywhere outside the box. If you want to verify the column matches, you can click the dropdown to check the header list.

NOTE: *Asterisks (*) in the import file are stripped out during the import process in RateManager, so you will not see them in the edit data box. Likewise, if you manually enter your data into the edit data box, there is no need to put an (*) in the wildcard field. Simply leave the field blank.*

7. Clicking **No** automatically matches the column headings to the variable(s) and variable criteria without viewing the import file data. This saves time when trying to import a large table.

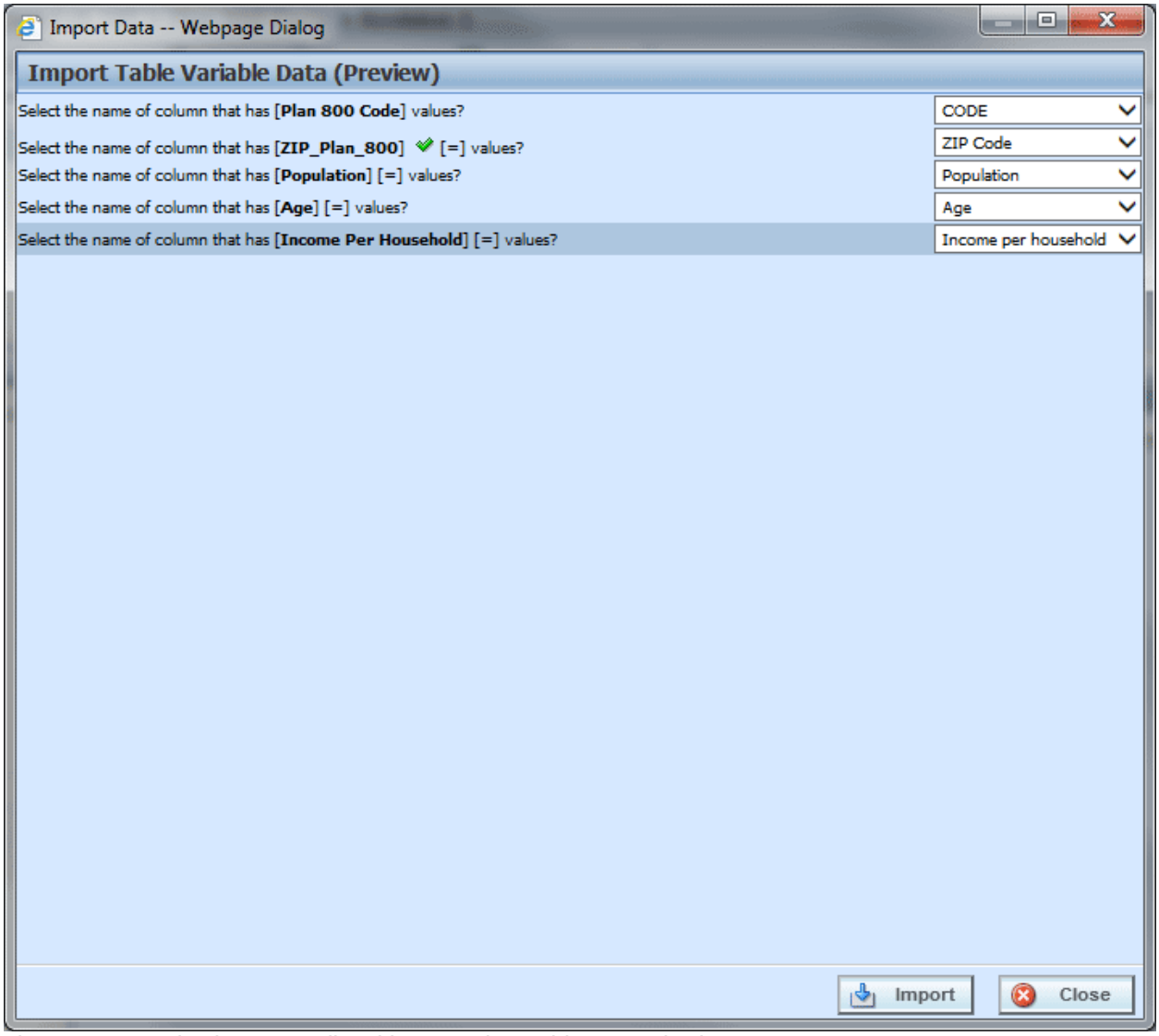


Figure 191 Previewing Data File with Manual Matching Required

8. To manually match columns to variable(s) and variable criteria, select the drop down next to a variable or criteria and choose the column from the import file that contains data for that item. When you are finished matching columns, select **Import**.

   If you do not have a match for a column you need, click **Close** to close the import process without bringing in any data.

   If you want to add or remove a column, you will need to re-import the data.

9. After columns have been matched, click Import. A success or fail message is displayed.

   A common reason for a file failing to import is the data type not matching. For example, if you have criteria that are dates and you have matched this to a string type column, then the import will fail. Failure messages will inform you what has failed and why.

10. If the table variable already contains data, you will be asked if you would like to delete the data first.

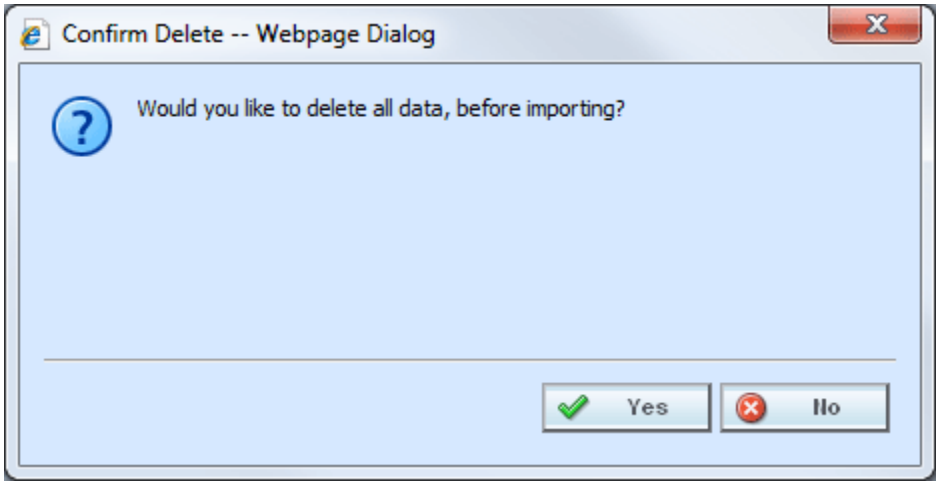


Figure 192 Confirming Delete All Data

11. Select **Ok** to have existing data in the table variable deleted before the new data is imported. Select **Cancel** to leave the existing data in the table variable and have the new data added at the end of the table.

12. When finished, a confirmation box is displayed, telling you how many rows were imported.

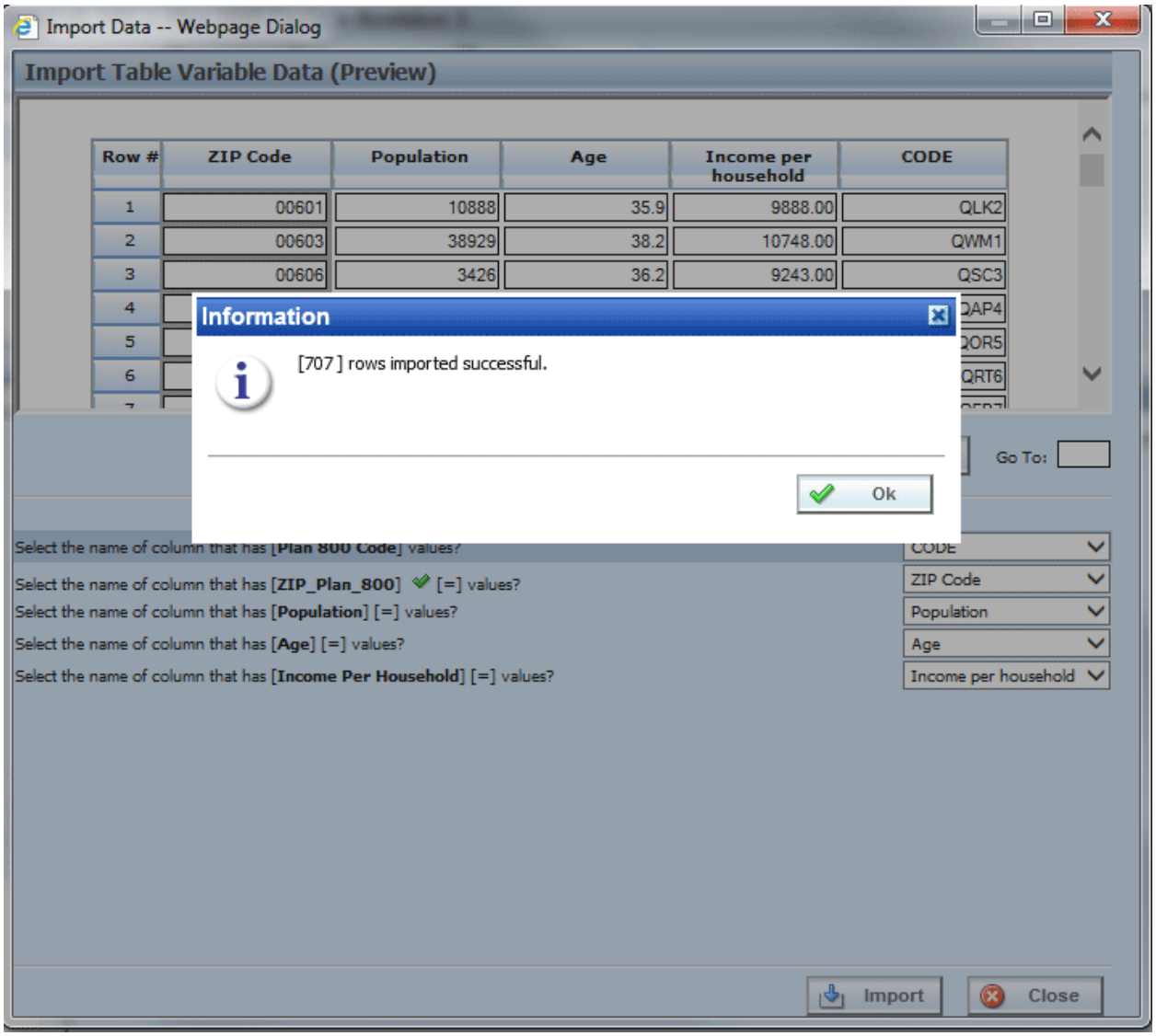


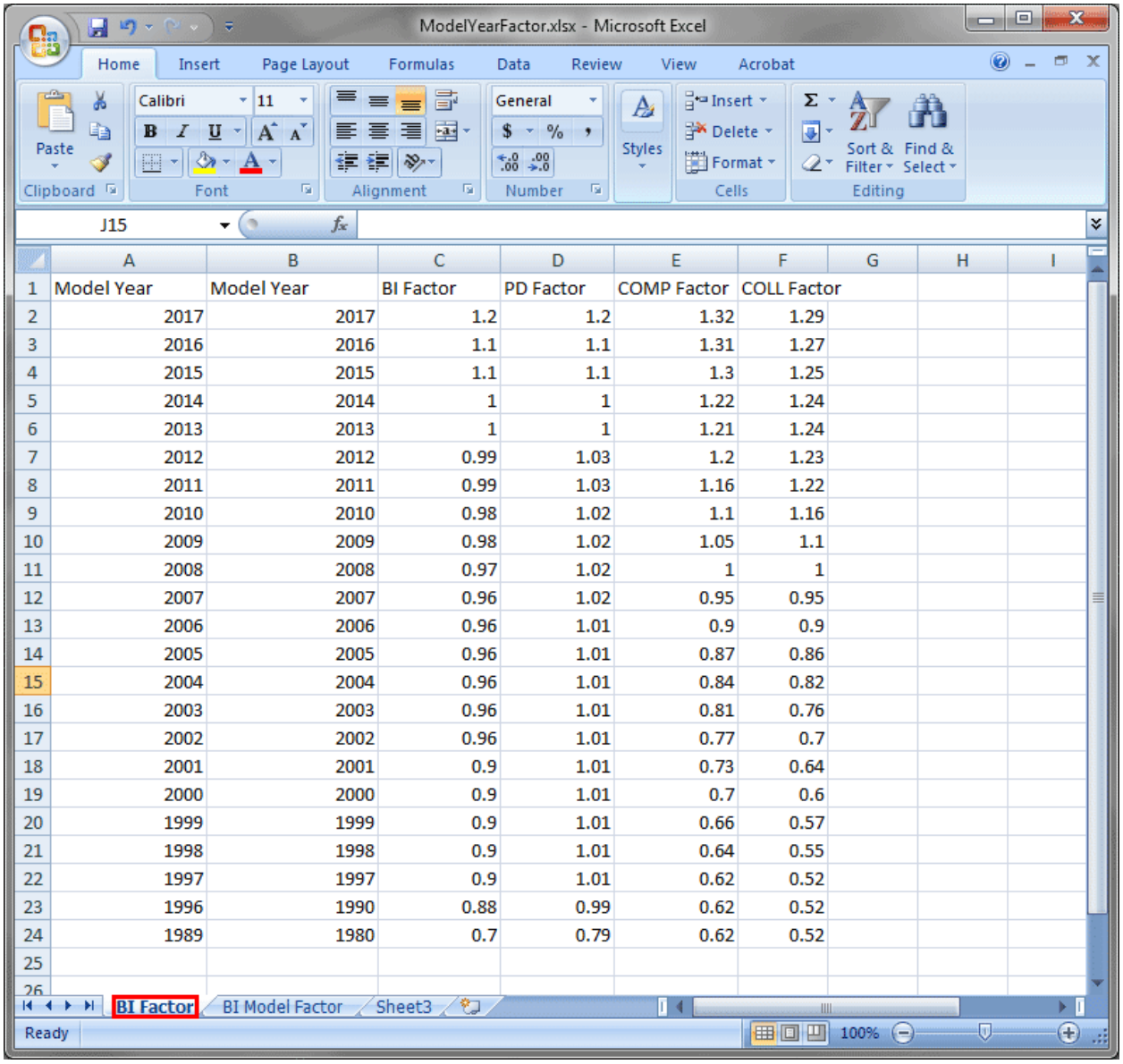
Figure 193 Successful Import Message

Working with MS Excel Files

MS Excel (.xlsx) files are created in Microsoft Excel. Requirements for importing Excel files are:

- The file must be saved in MS Excel format using an .xlsx extension.
- The file is a workbook. Each tab in the workbook is a worksheet.
- The workbook name can be any name you choose. When importing, the default workbook name is the table variable name. The workbook can be selected from either your local machine or network, or from the workfiles server.
- Once the workbook has been imported, it is saved to the workfiles location and can be selected again.
- The worksheet tab must be named. If the worksheet is not named, the import will fail. The default name for the worksheet is the table variable name.
- Excel files can be imported in bulk or one at a time.

- Only one worksheet can be mapped to a workbook. If you need to use a second worksheet, you must save the workbook using another workbook name and import the worksheet with the updated workbook name. When the workbook is imported, the mapping is saved for the named worksheet.



	A	B	C	D	E	F	G	H	I
1	Model Year	Model Year	BI Factor	PD Factor	COMP Factor	COLL Factor			
2	2017	2017	1.2	1.2	1.32	1.29			
3	2016	2016	1.1	1.1	1.31	1.27			
4	2015	2015	1.1	1.1	1.3	1.25			
5	2014	2014	1	1	1.22	1.24			
6	2013	2013	1	1	1.21	1.24			
7	2012	2012	0.99	1.03	1.2	1.23			
8	2011	2011	0.99	1.03	1.16	1.22			
9	2010	2010	0.98	1.02	1.1	1.16			
10	2009	2009	0.98	1.02	1.05	1.1			
11	2008	2008	0.97	1.02	1	1			
12	2007	2007	0.96	1.02	0.95	0.95			
13	2006	2006	0.96	1.01	0.9	0.9			
14	2005	2005	0.96	1.01	0.87	0.86			
15	2004	2004	0.96	1.01	0.84	0.82			
16	2003	2003	0.96	1.01	0.81	0.76			
17	2002	2002	0.96	1.01	0.77	0.7			
18	2001	2001	0.9	1.01	0.73	0.64			
19	2000	2000	0.9	1.01	0.7	0.6			
20	1999	1999	0.9	1.01	0.66	0.57			
21	1998	1998	0.9	1.01	0.64	0.55			
22	1997	1997	0.9	1.01	0.62	0.52			
23	1996	1990	0.88	0.99	0.62	0.52			
24	1989	1980	0.7	0.79	0.62	0.52			
25									
26									

Figure 194 Importing Table Data from an Excel File

For example, in Figure 194 Importing Table Data from an Excel File, the workbook ModelYearFactor has two worksheet tabs; BI Factor and BI Model Factor. If a table variable has been created called BI Factor and the data imported was from the worksheet ModelYearFactor, then the data from the worksheet BI Model Factor cannot be imported using the same workbook. The workbook must be saved under a different name and that will allow for the BI Model Factor worksheet to be imported.

Workbooks are saved to the workfiles location with the file name and a unique identifier. Workbooks are saved with the original mapping used. If this file is no longer needed, system administrators can delete files from the workfiles location.

MS Excel File Import Data File Options

There are multiple options for importing an Excel file.

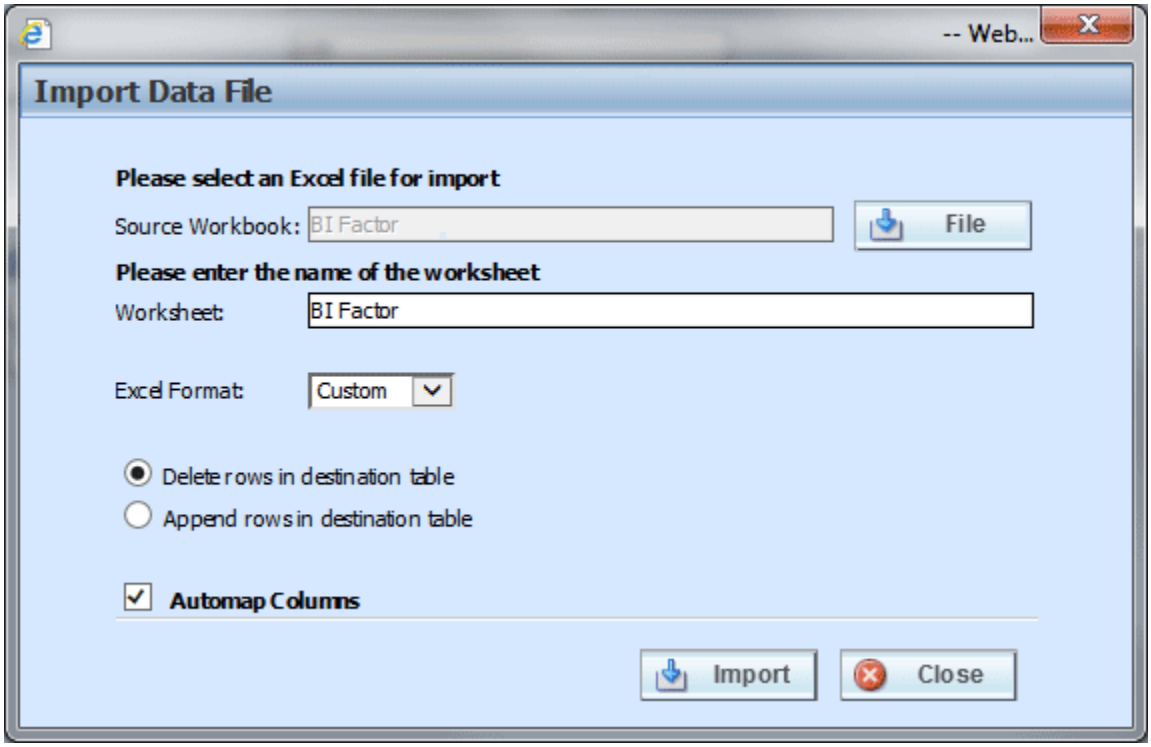


Figure 195 Import Excel File Options

- **Source Workbook:** The Excel file to be used. The file must use an .xlsx extension. If the right format is not found, an error message is returned. The default is the table variable name. Click File to select a file on your local machine or network, or to select a file from the workfiles location.
- **Worksheet:** The name of the tab to be imported. The default is the table variable name. You can update this field manually.
- **Excel Format:** There are two options:
  - **Custom** – Custom files are files that have not been imported before. These are the files on your local machine or network.
  - **Insbridge** – Insbridge files are files that have been imported and are saved on the workfiles location. These files have mappings associated to them.
- **Delete rows in destination table** – select this option to delete any data in the table. This is the default.
- **Append rows in the destination tabl**e – adds the data to the end of the table.
- **Automap Columns** – automatically maps table variables and criteria to the columns in the data. This option is checked by default. If you want to manually map the columns, uncheck the automap.

Importing a MS Excel File in to a Table Variable

When importing Microsoft Excel files, there are a few options, depending on permissions. In this example, a table variable with ten selection criteria will be imported by a user with table job permissions.

1. Navigate to the **Edit Variable** tab for the table variable from where you want to import a table. Make sure you have any linked variables and selection criteria already entered. **Save** your entry prior to importing.

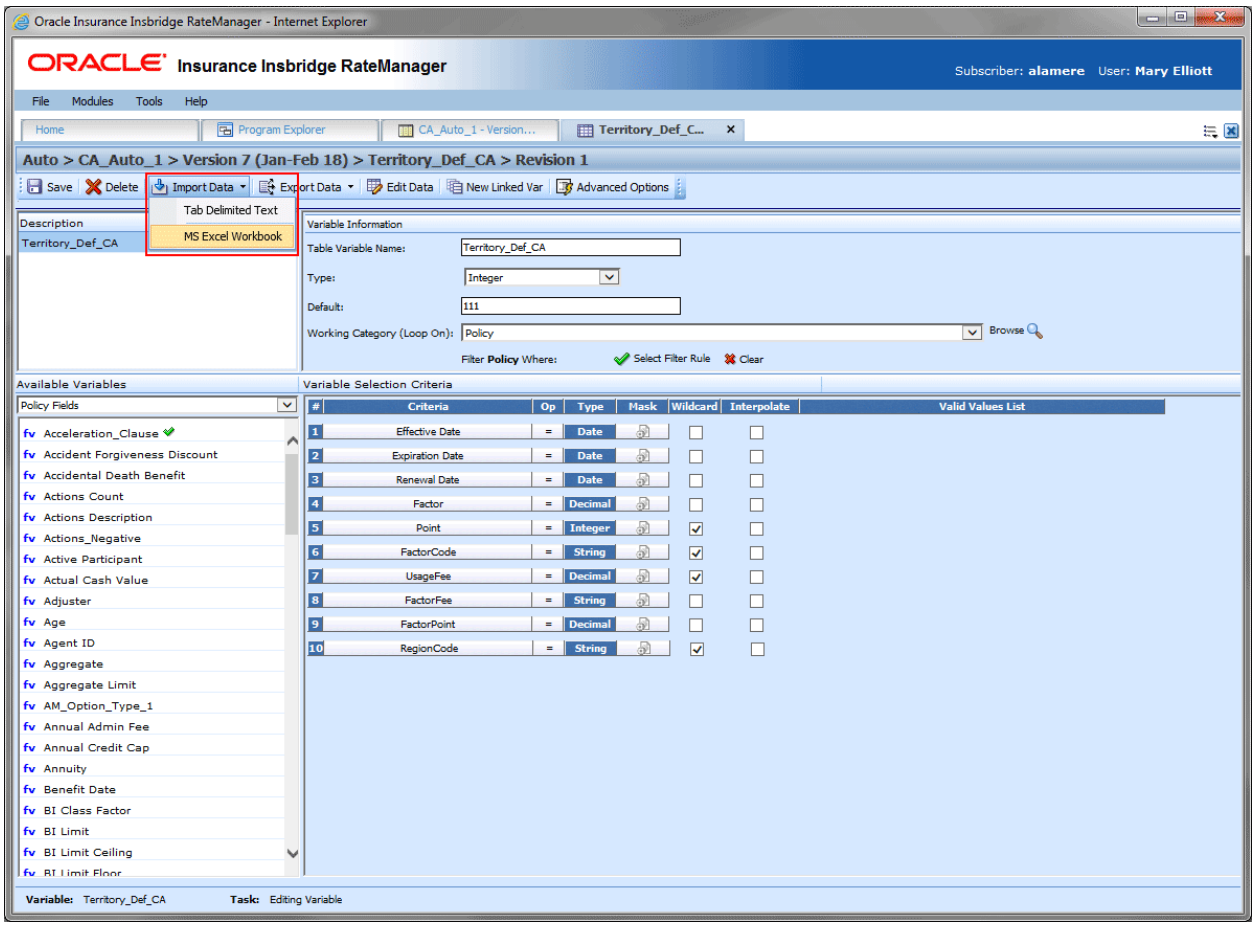


Figure 196 Selecting Data File

2. Click **Import Data** and select MS Excel Workbook.

3. The import data file defaults to the name of the table variable. To select a new file, click File.

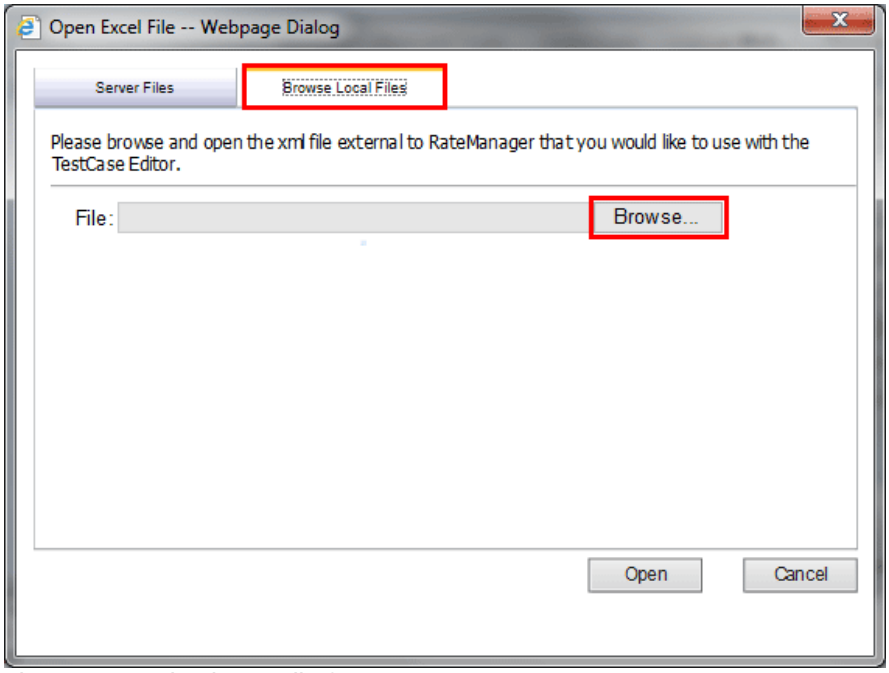


Figure 197 Selecting a File for Import

4. You can select to browse server files or local files. Click the Browse Local Files tab.

5. Select the data file you want to import from your local machine or network. Once the file has been selected, click Open to begin the import process. The location path of the file is entered in the selection field.

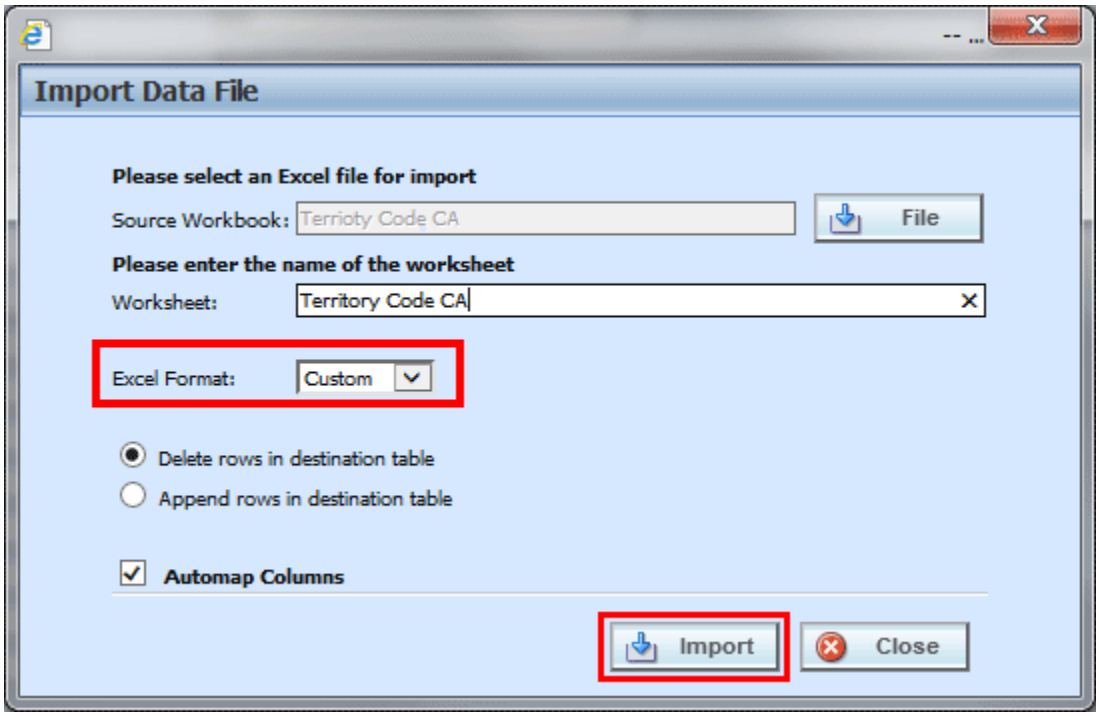


Figure 198 Import Table Data

6. The workbook name is in the Source Workbook Field. This field can only be changed by selecting another file.

7. The Worksheet name defaults to the name of the table. Make sure the name is the same as the sheet in the Excel file.

NOTE: *If the worksheet name does not match or the worksheet in the Excel file is not named, the import will fail*

8. For a new file, the Excel format must be **Custom**. If Insbridge is selected, the import rows will begin at row 10.

NOTE: *If your imported data begins at row 10, skipping rows 1-9, the Insbridge format has been used. If this is incorrect, you must change the format to Custom and re-import*

9. The Delete rows in destination table or Append rows in destination table options do not apply for a new file. If this is a re-import and you want to add to or deleted previous data, select the appropriate option.

10. The automap feature is checked by default. If you do not want the fields to automatically map, uncheck the option. Auto mapped fields can be changed.

11. Click **Import** to upload the file.

12. You will be asked if you would like to preview your data.

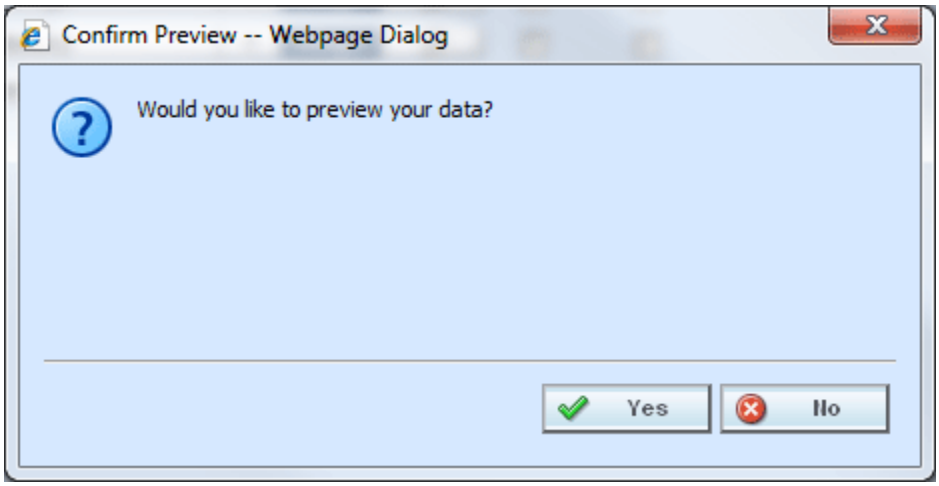


Figure 199 Confirming Preview of Data

13. Clicking **Yes** also matches the column headings to the variable(s) and variable criteria automatically and displays the import file. This allows you to verify the matches are correct.

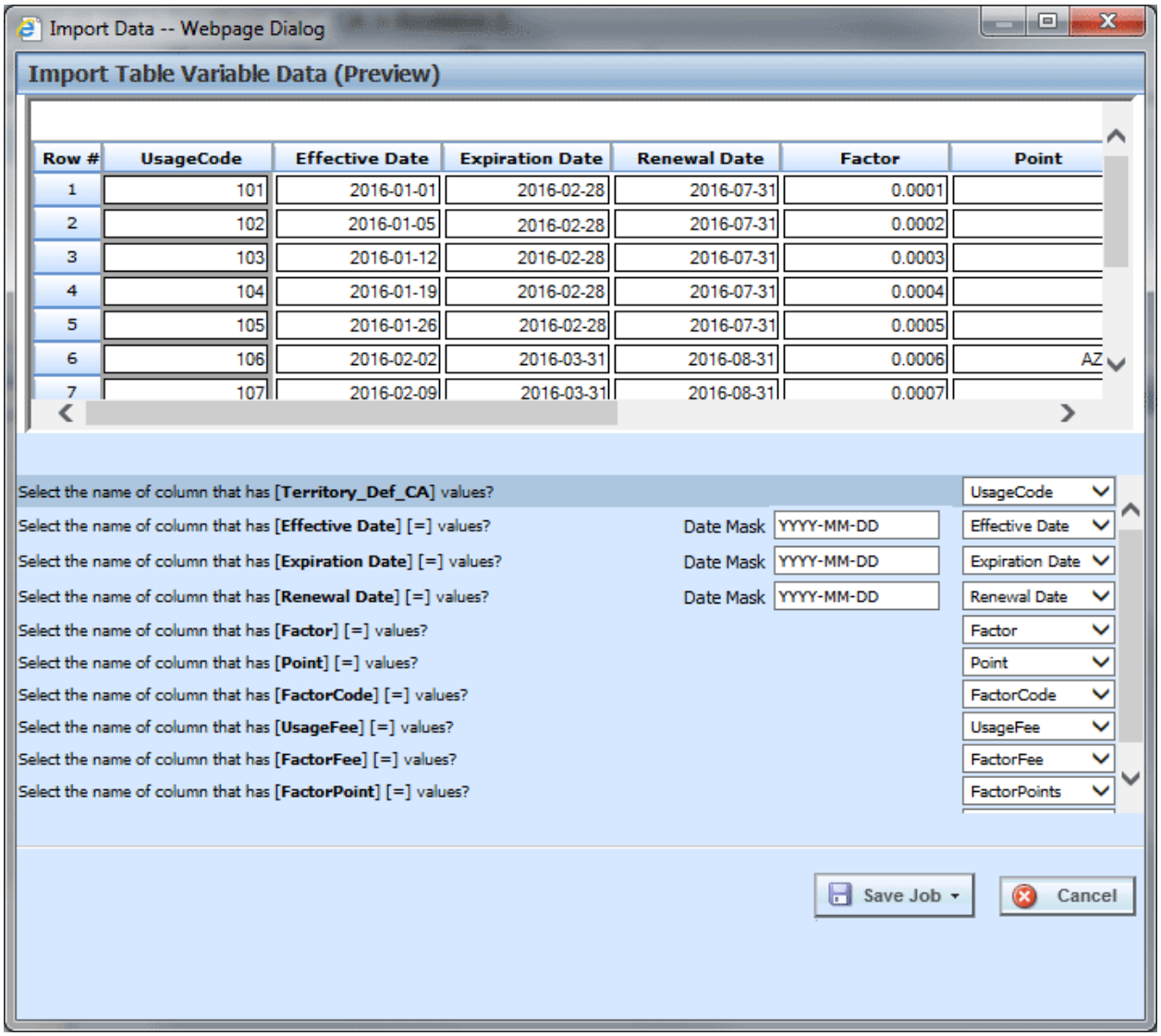


Figure 200 Previewing Data File with Automatically Matched Columns

The import window is split into two parts. The top portion displays the import file as you created it. The bottom half lists the criteria and variable(s) you created within RateManager. If the column headers, the criteria and table variables names match exactly, the columns will be matched automatically. If there is a difference in the name, you will have to match up each element with the correct column in the import file manually.

NOTE: *For auto mapping, column header names, criteria and table variable names must match exactly, including capitalization*

If the table contains multiple rows, you can use **Next 50** and **Previous 50** to navigate through the table. You also can go directly to a row by entering the row number in the **Go To box** and clicking anywhere outside the box.

NOTE: *Asterisks (*) in the import file are stripped out during the import process in RateManager, so you will not see them in the edit data box. Likewise, if you manually enter your data into the edit data box, there is no need to put an (*) in the wildcard field. Simply leave the field blank.*

13. Clicking **No** automatically matches the column headings to the variable(s) and variable criteria without viewing the import file data. This saves time when trying to import a large table.

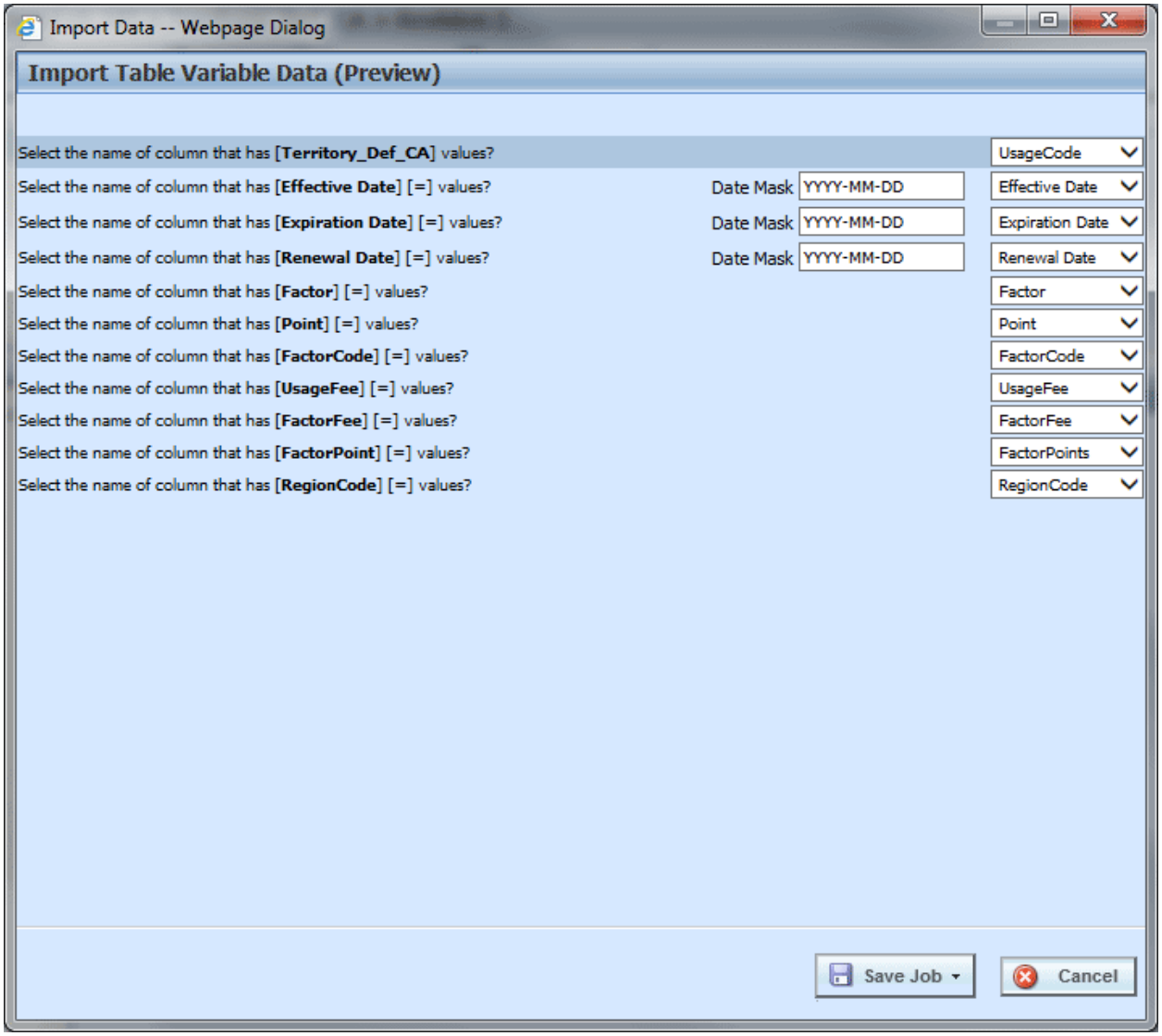


Figure 201 Previewing Data File with Manual Matching Required

14. To manually match columns to variable(s) and variable criteria, select the drop down next to a variable or criteria and choose the column from the import file that contains data for that item.

    If you do not have a match for a column you need, click **Cancel** to close the import process without bringing in any data.

    If you want to add or remove a column, you will need to re-import the data.

15. When you are finished matching columns, select **Save Job**.

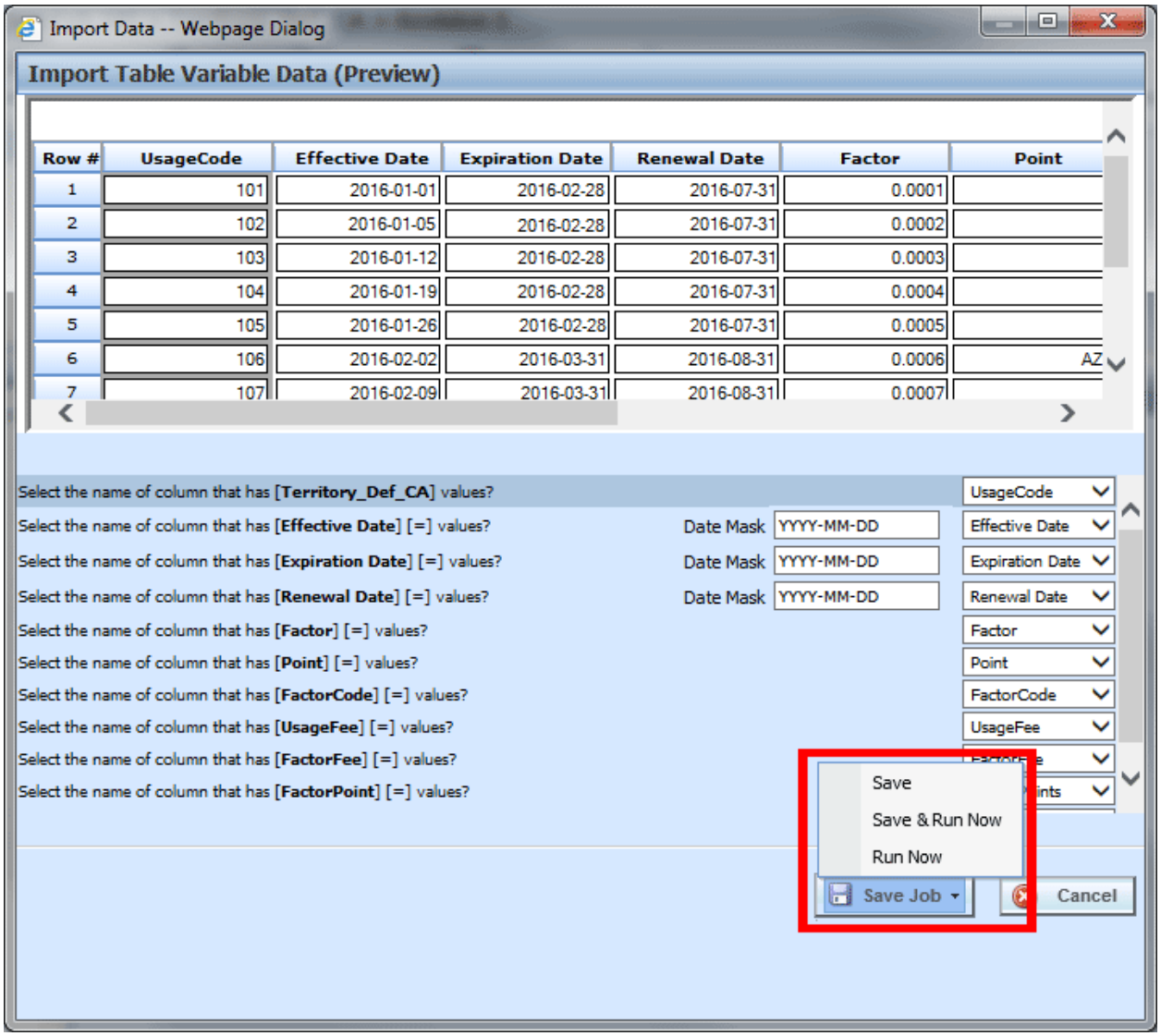


Figure 202 Import File Job Options

Options for Importing Data

If this is the data you want to import, click Save Job and select the option you want. The options presented depend on the permissions you have.

- **Save** – saves the import to a job listed on the Job Management tab. If no jobs have been created, the import cannot be saved. You can still run the job with Run Now.
- **Save and Run** – saves the import to a job listed on the Job Management tab and then runs the job. If no jobs have been created, the import cannot be saved. If you do not have Job Management permissions, this option is not displayed.
- **Run Now** – runs the import job immediately.

Once the job has been successful, an email is sent to the users on the table job.

16. If you opt to Save, you can select the job where you want to save the export to. A success or fail message is displayed.

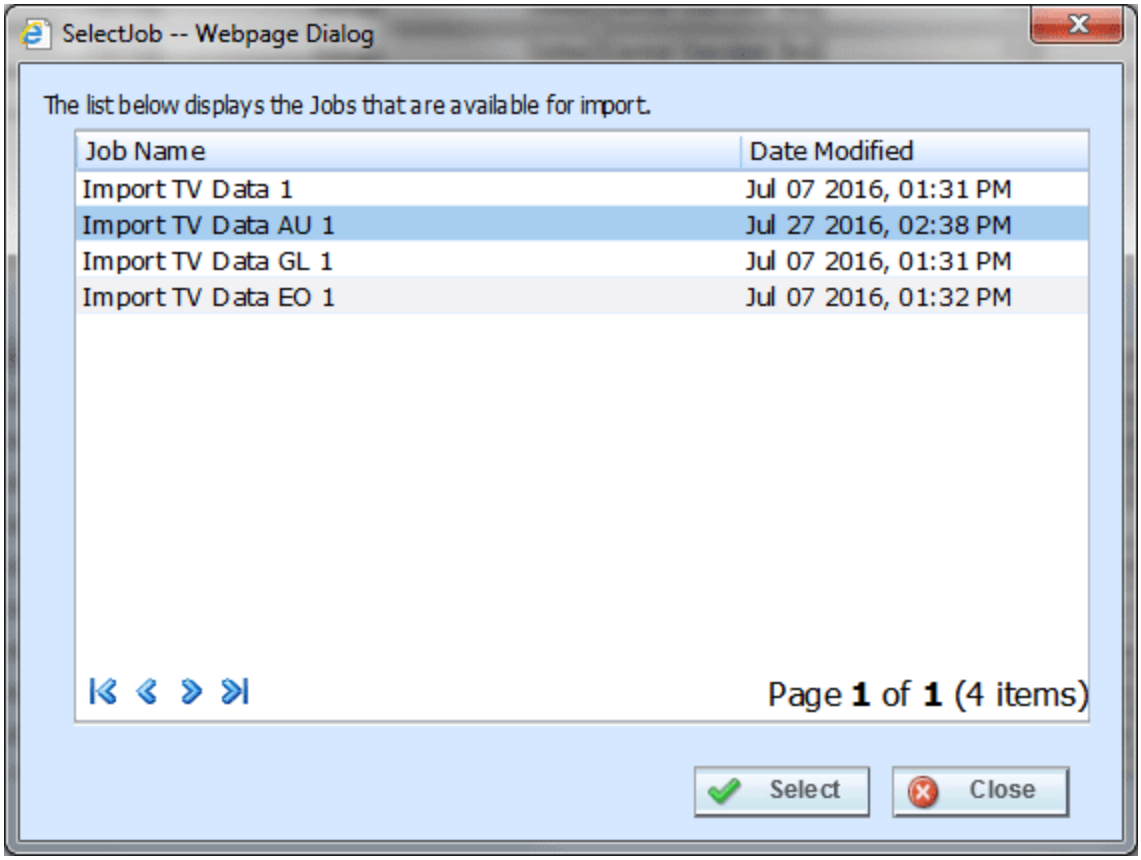


Figure 203 Selecting a Job for Import

17. After the imports have been saved, users with permissions can view, edit and start the job on the Table Job Management screen.

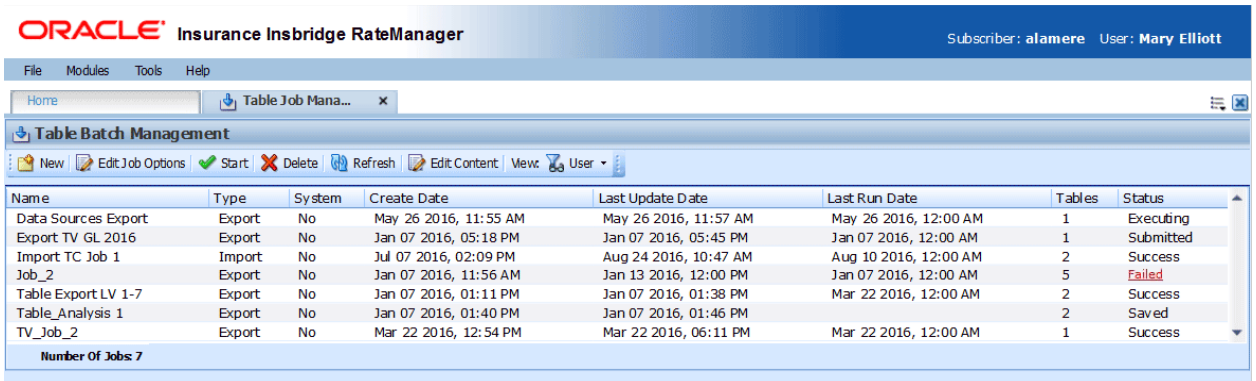


Figure 204 Table Job Management

Importing Data Immediately

Two options allow for the immediate import of table data:

- **Save and Run** – saves the import to a job listed on the Job Management tab and then runs the job. If no jobs have been created, the import cannot be saved. If you do not have Job Management permissions, this option is not displayed.
- **Run Now** – runs the import job immediately.

If one of these options is selected, the job is run immediately. Once the job has been successful, an email is sent to the users on the table job.

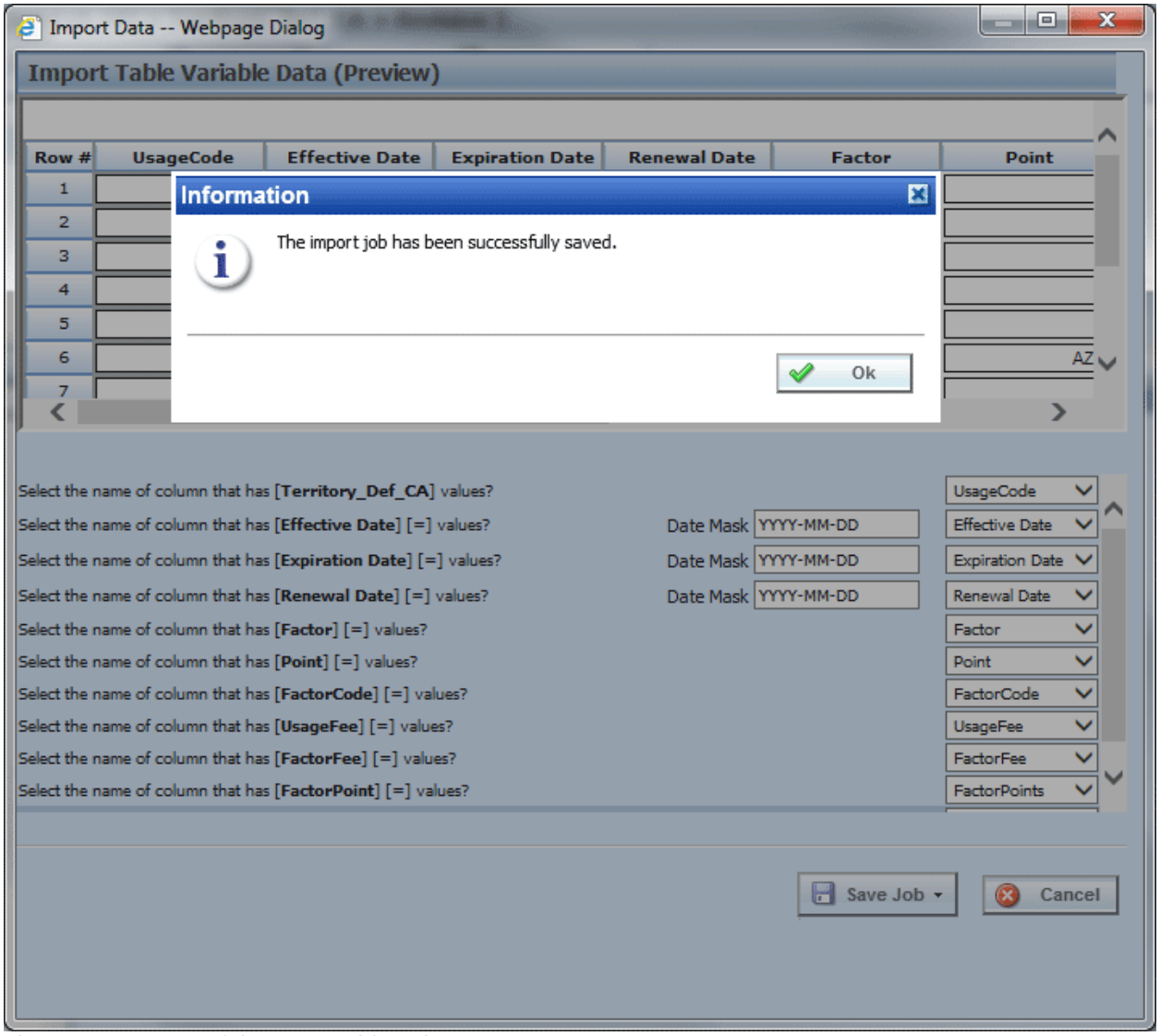


Figure 205 Successful Import Table Job

Using a Date Mask

If enabled, a date mask (format) for 'Date' data types is displayed. This date mask allows to you define the format of the date you are importing. The date mask identifies the format that the date is currently in before it is imported. When the data is imported in, it will be converted. The format depends on the level of the variable. Local variables follow the format defined at the program level. Global or universal variables are defined at the product level.

Date masks can be set at the product level and at the program level. At the product level, the date mask will apply to all programs in that product. At the program level, the date mask will apply to all versions of the program. A program level date mask will override a product date mask. Wherever the date format is defined, either at the product or program level, is the format that will be used when rating.

Importing Date Data

1. When you are ready to import your table, click **Import Data** on the navigation bar to open the Import Data File screen.

2. When you are asked if you would like to preview your data, click **Yes** to see your data or **No** if you do not want to see your data. Selecting NO is helpful if there is a large amount of data.

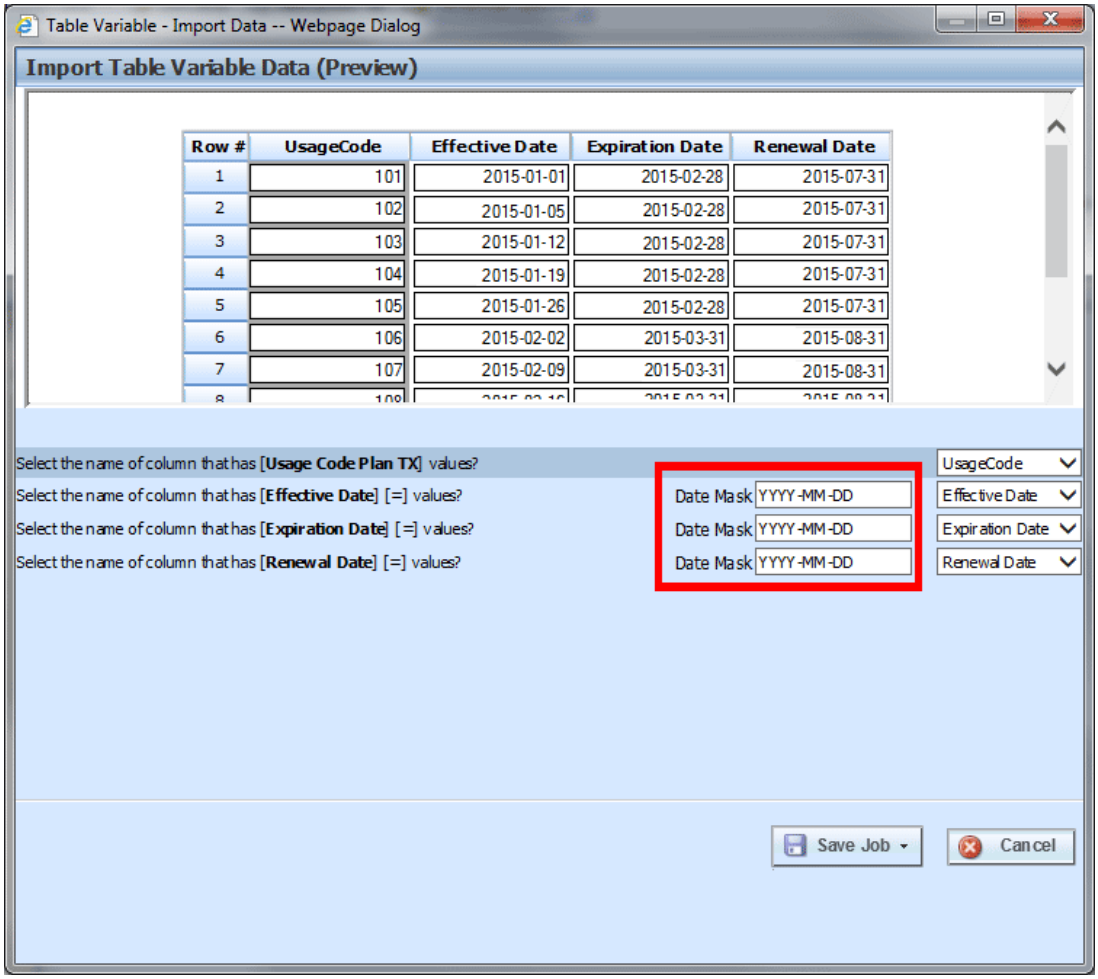


Figure 206 Previewing Data File with a Date Mask

3. Match the variables and criteria on the left side to the corresponding column headings on the right side.

4. Enter the date mask (format) of the data you are importing.

5. When you are finished, select **Import**.

6. If the table variable already contains data, you will be asked if you would like to delete the data first. Select **Ok** to have existing data in the table variable deleted before the new data is imported. Select **Cancel** to leave the existing data in the table variable and have the new data added at the end of the table.

7. When finished, a confirmation box will display, telling you how many rows were imported. If there is an error, please correct the error and try again.

Editing Imported Table Data

When you enter the table in RateManager you will see the date format that the program has.

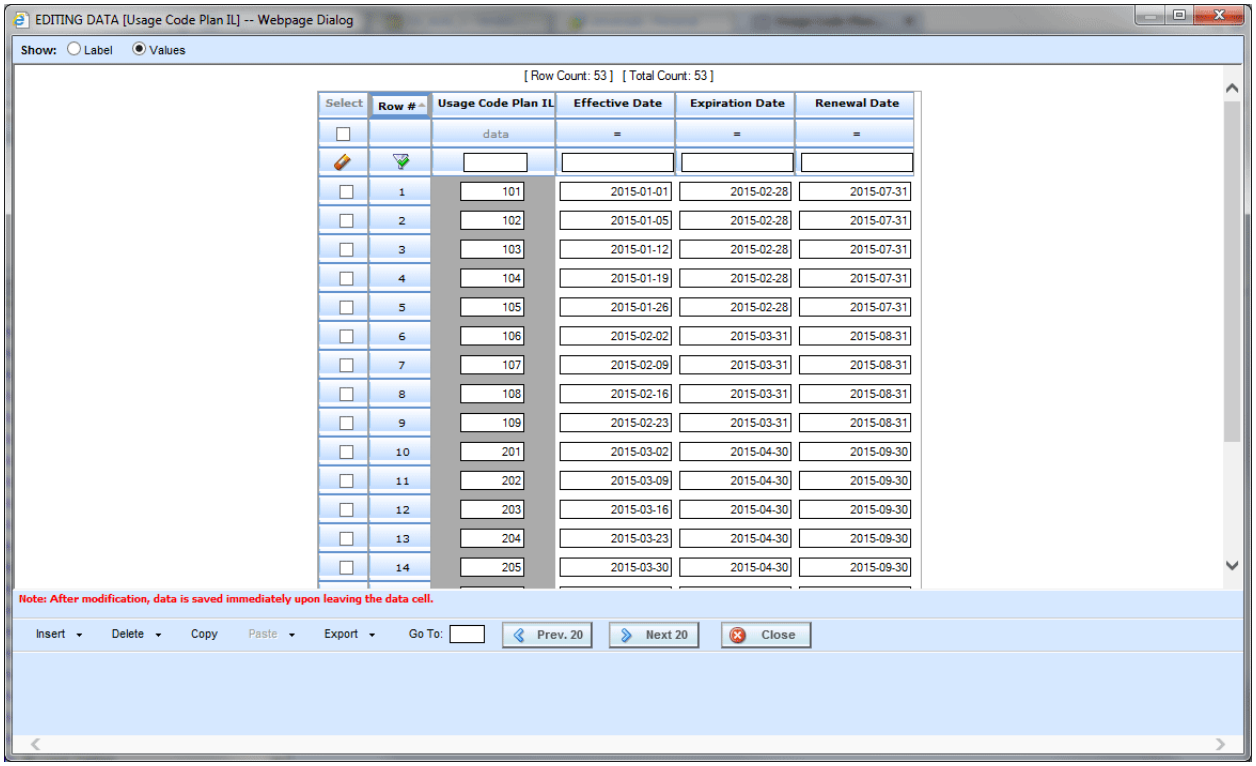


Figure 207 Imported Table Data

The date format that you imported will be changed to the date format that is on the program.

Notes on Date Masks

- The date mask applies only to the data you are importing.
- Date formats must match exactly, including the separators being used.

- Day and month formats must be two digits. For example January 7 must have a zero proceeding the month (01) and proceeding the day (07).
- Months must be represented in numeric form. 1-Jan-2012 will not be imported. 01-01-2012 would be imported.
- If you edit a row or enter additional rows, the date mask will not apply. Any edits or additions completed within RateManager should be in the proper format. If you have extensive updates or additions, it is recommended that you edit the table outside of RateManager and re-import.
- If a date mask field is present, you must make an entry.
- If the system administrators have not enabled this feature, this field will not be visible.

Creating a Table without Importing Data

If you do not import a table, you can create one using the Edit Data screen. After you have created your table variable, click Edit Data to open the Edit Data screen. There will be columns for the table variable(s) you are creating and the criteria you have selected. Click **Insert** and then select either **Above Row** or **Below Row** to insert the first row. Enter any data and continue adding rows until you are finished. When you have finished entering your data, click **Close** to return the **Edit Variable** tab.

EXPORT DATA

Exporting a table places all the data from a table variables data table into a tab-delimited text file or a MS Excel workbook. There are three locations where a table variable can be exported:

- The edit tab of any selected table variable has an Export Data option in the top bar menu that contains a drop down where you can select to export data to a tab delimited text or to an MS Excel workbook. This option is open to all users with product permissions and allows to you export the exact revision you need.

- The program details tab of any selected program version can display the global or local table variables for the program. All table variables have a right click menu with the Table Management option. From there, you can select to export data either to a tab delimited text or to an MS Excel workbook. This option is open to all users with product permissions.

- The universal tab of any project can display the table variables for the project. All table variables have a right click menu with the Table Management option. From there, you can select to export data either to a tab delimited text or to an MS Excel workbook. This option is open to all users with Universal permissions. If you do not have universal permissions, you may receive an error.

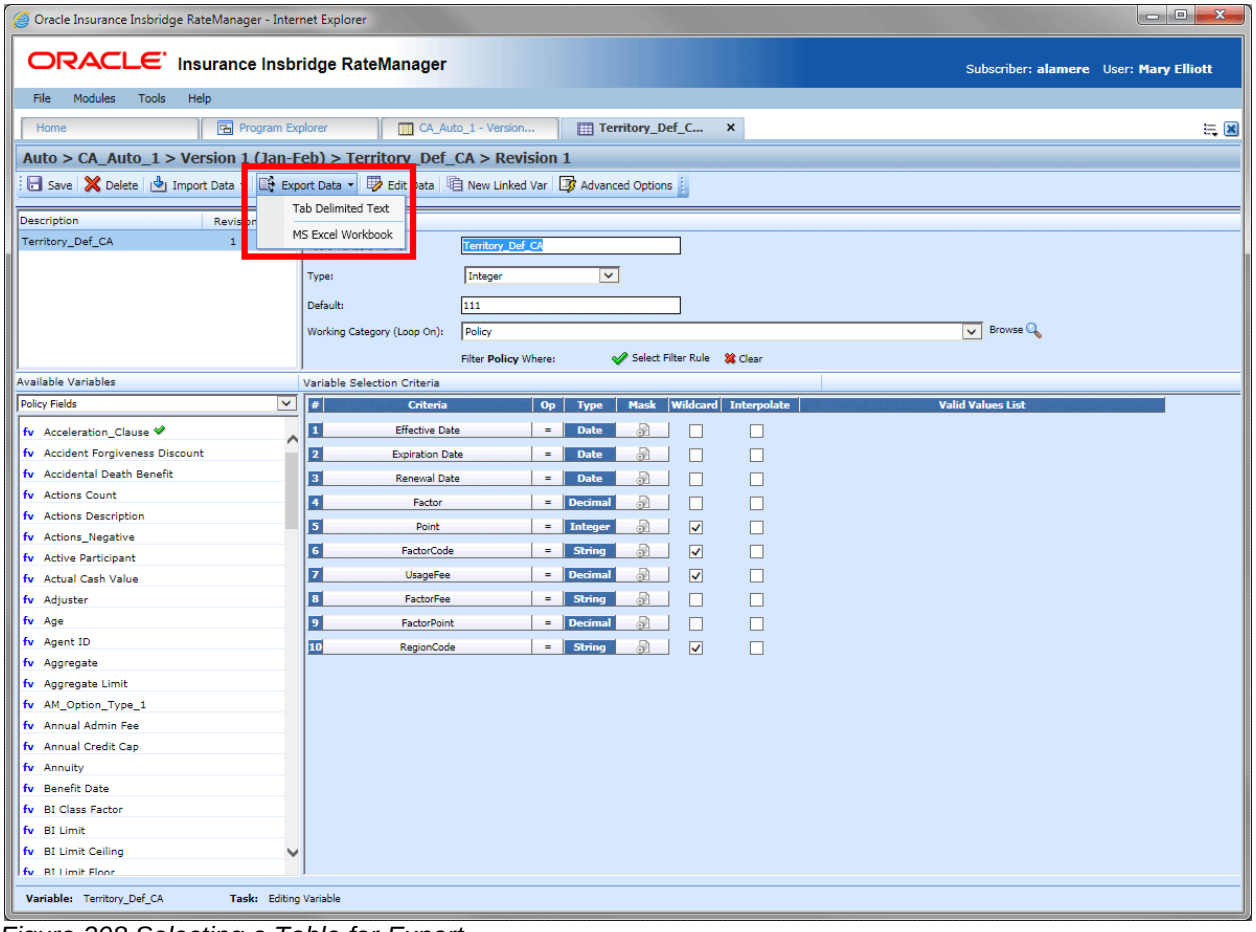


Figure 208 Selecting a Table for Export

There are two ways to export data:

- **Export Data (Tab-Delimited):** Exports table data into a tab-delimited text file. You can then make changes to the file and import the data into the same variable or a different variable. This option is for locked and unlocked table variables only.
- **Export Data (MS Excel Workbook):** Exports table data into a Microsoft Excel workbook where the data can be viewed. Files cannot be brought back in to RateManager at this time. This option has setup requirements, permissions that need to be assigned and is for locked and unlocked table variables only.

Export Data and Wildcards

When wildcard null values are part of the data set, no value is placed in the exported data field. This may present issues when editing or reviewing data. If you want an asterisk to be placed where a wildcard null value is located, then select the Include asterisk for Wildcard NULL value option.

Tab-Delimited Table Data

For a tab-delimited export, no Export Data File screen is presented. The download is automatically done. Only the browser option to Open, Save or Cancel is presented. The asterisk is included only if the *Table Management Export Include Asterisk for Wildcard* option is set to True in Preferences. Preference settings are available to administrators only.

If the *Table Management Export Include Asterisk for Wildcard* option is set to False in Preferences, no asterisks are added to the exported data. The null fields are blank.

Excel Table Data

For Excel table data exports, the asterisk is included if the “Include asterisk for Wildcard NULL values” is checked on the Export Data File screen. This option is automatically checked if the *Table Management Export Include Asterisk for Wildcard* option is set to True in Preferences. It can be unchecked if needed. If the option is set to false, then the box is not automatically checked but can be checked at the time of export. Preference settings are available to administrators only.

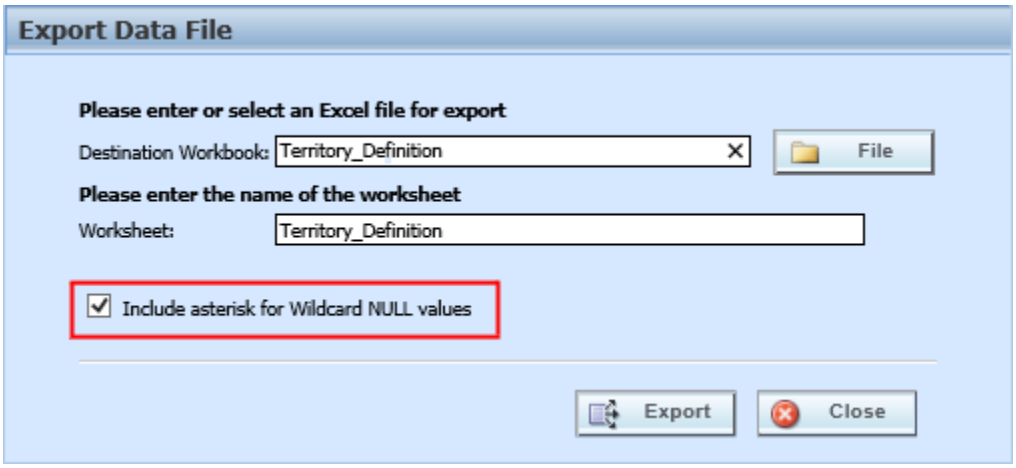
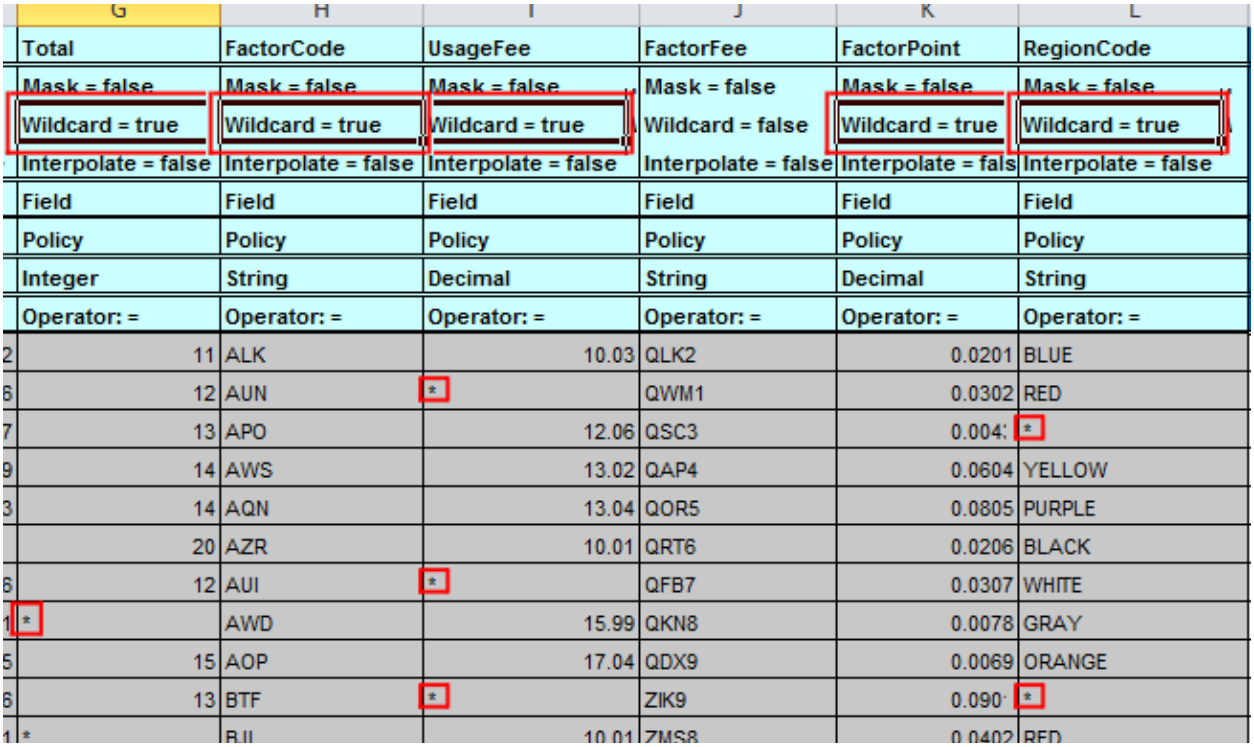


Figure 209 Export Data Including Wildcard Null Values



	G	H	I	J	K	L
	Total	FactorCode	UsageFee	FactorFee	FactorPoint	RegionCode
	Mask = false	Mask = false	Mask = false	Mask = false	Mask = false	Mask = false
	Wildcard = true	Wildcard = true	Wildcard = true	Wildcard = false	Wildcard = true	Wildcard = true
	Interpolate = false	Interpolate = false	Interpolate = false	Interpolate = false	Interpolate = fals	Interpolate = false
	Field	Field	Field	Field	Field	Field
	Policy	Policy	Policy	Policy	Policy	Policy
	Integer	String	Decimal	String	Decimal	String
	Operator: =	Operator: =	Operator: =	Operator: =	Operator: =	Operator: =
2	11	ALK	10.03	QLK2	0.0201	BLUE
6	12	AUN	*	QWM1	0.0302	RED
7	13	APO	12.06	QSC3	0.004	*
9	14	AWS	13.02	QAP4	0.0604	YELLOW
3	14	AQN	13.04	QOR5	0.0805	PURPLE
	20	AZR	10.01	QRT6	0.0206	BLACK
6	12	AUI	*	QFB7	0.0307	WHITE
1	*	AWD	15.99	QKN8	0.0078	GRAY
5	15	AOP	17.04	QDX9	0.0069	ORANGE
6	13	BTF	*	ZIK9	0.090	*
1	*	BJI	10.01	ZMS8	0.0402	RED

Figure 210 Include Asterisks for Wildcard Null Values

Export Data Tab-Delimited

Exporting a table in to a tab-delimited text file does not require any set up or permissions. If you have permissions to the product you can export a table variable in to a tab-delimited file. This option allows you to make changes to the file and then import the data back in to the same variable or a different variable.

NOTE: *Exported tab-delimited files may contain RateManager added information such as default values, data type, and operators. This information must be deleted from the file before being brought back in to RateManager.*

If you are exporting a table from the Program Details or Universal tab, the active revision is displayed on the listing. If you want to export a table variable revision that is not the active revision, you need to show/hide revisions and select the revision you want to export.

1. Navigate to the **Edit Variable** tab for the table variable from where you want to export data. You also can export the data on the Program Details tab by right clicking the table variable and selecting **Export Data (Tab-Delimited)** from the menu.

2. Click Export Data. You will be prompted by your browser to confirm downloading the file.

3. Click Save to save to the default area of your local hard drive, or **Save As** to save the file to a designated place on your local or network hard drive or **Open** to open the file with your default text editor.

4. If you chose to Save As the file, you will be asked to choose a location to save the file to.

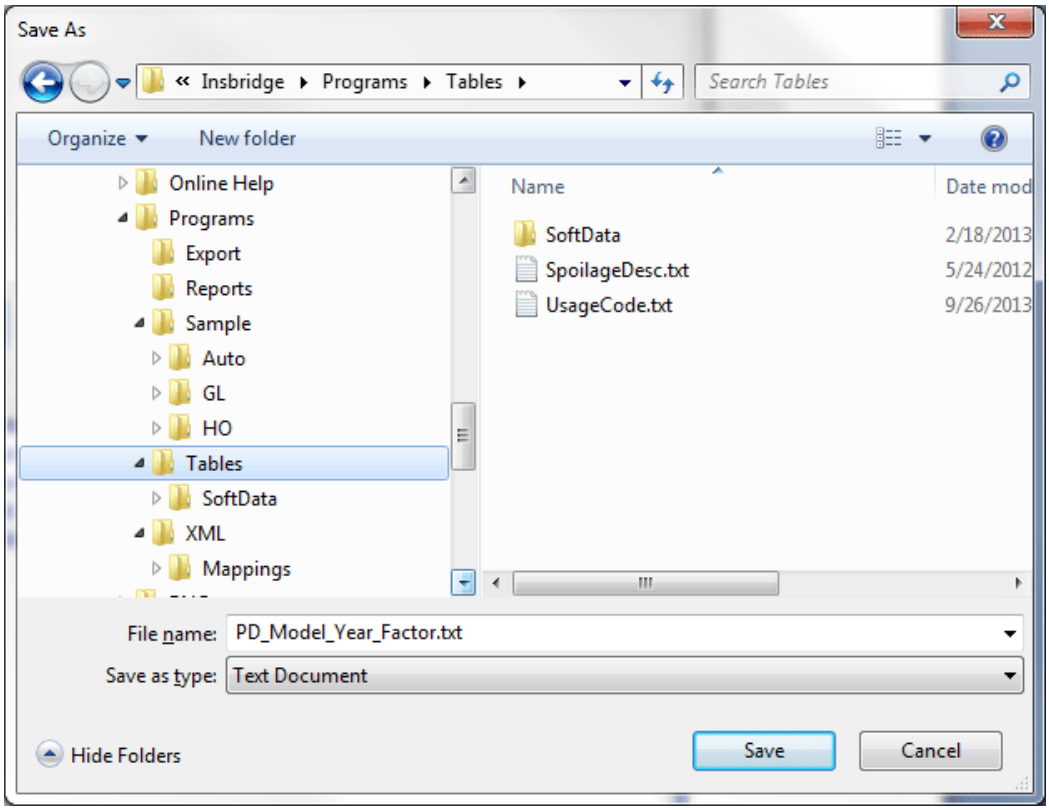


Figure 211 Saving a Table

5. Select the folder where you want to save the file and enter in a name for the file. By default, the file will be saved as the variable name. Click **Save**, the file will be saved to the location you specified. When the file has finished downloading, you will have the option to:

    a. **OPEN:** this opens the file in your default text editor.
    b. **OPEN FOLDER**: this opens your computers explorer.
    c. **Close**: this closes the dialog box.

    You can now edit the file with the text editor of your choice. Edited tab-delimited files can be re-imported back in to RateManager.

Exported Tab Delimited Data and Unicode

RateManager allows for Unicode characters to be imported and exported from tables. Exported tables that have Unicode characters in it may not display properly in some editing applications.

If you attempt to open an exported text file, .txt, with an editing application, such as Microsoft Excel, and the Unicode characters are not present, you may have to open the editing application first and then choose Open (file). This method will prompt you to select the File Origin (encoding). If you select UTF-8, the data will be displayed properly.

You can open the exported text file using a text editor such as Notepad and the do a Save or a Save As (save with Encoding of UTF-8). Next open your editing application and open the text file. The values should be correctly displayed.

Export Data to MS Excel Workbook

Exporting data to a Microsoft Excel workbook has requirements, setup, and permissions that need to be assigned. Details on the requirements, setup, and permissions can be found in the Insbridge RateManager Admin User Guide under Table Job Management.

Table Job Management an option on the Tools→Setup menu where users with permissions manage table export jobs. Jobs are containers that can hold one or more tables for export. In order to export multiple tables at one time, a job must be created. Jobs are not required if you want to export one table immediately.

To Export to MS Excel

If you are exporting a table from the Program Details or Universal tab, the active revision is displayed on the listing. If you want to export a table variable revision that is not the active revision, you need to show/hide revisions and select the revision you want to export.

1. Select the table variable revision that you want to export.

2. Select to Export a table to MS Excel.

The Export Data File screen is displayed. The first step is to specify the Destination Workbook name and the Worksheet name. Both fields are required. The defaults are the name of the table. If you want to export a table variable that already has been exported, you can re-name the workbook. This creates a new file. If you have selected a linked variable, they are all included in the export. For more on the Export Data File, see

3. Export Data File.

   The “Include asterisk for Wildcard NULL values” option can be either checked or unchecked. See Export Data and Wildcards for more information.

NOTE: *You can also choose an existing file in your server if you want. Once you set up both fields you can click on export. Please see Insbridge RateManager Admin User Guide under Table Job Management.*

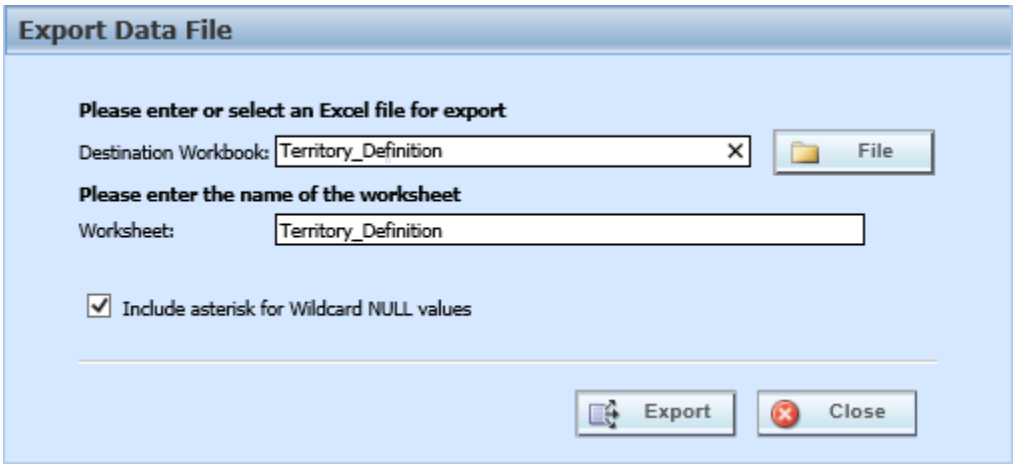


Figure 212 Export Data File

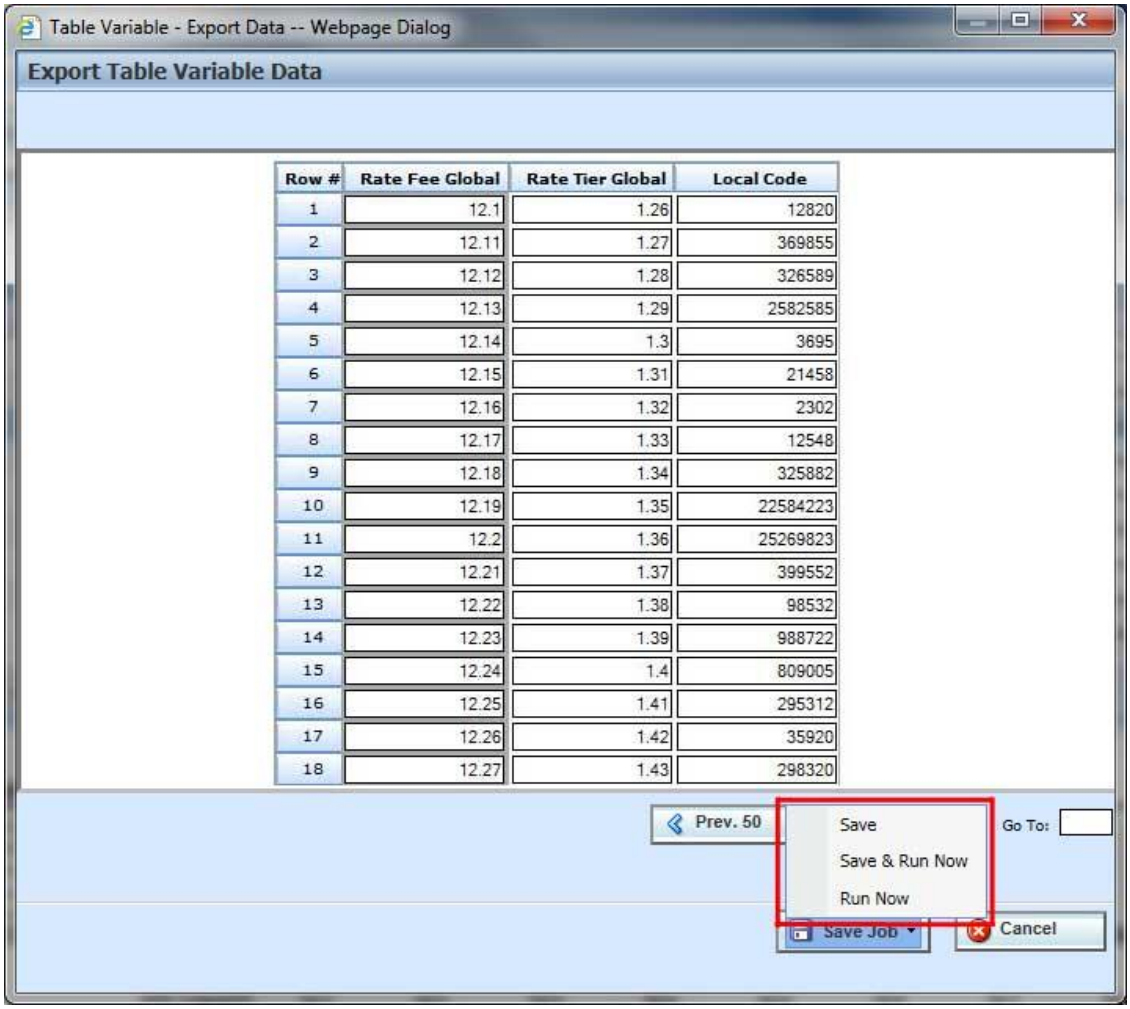


Figure 213 Options for Exporting Table Data

4. The Export Table Variable Data screen is next. The data can be reviewed but not edited. If this is the data you want to export, click Save Job and select the option you want. The options presented depend on the permissions you have.

- **Save** – saves the export to a job listed on the Job Management tab. If no jobs have been created, the export cannot be saved. You can still run the job with Run Now.

- **Save and Run** – saves the export to a job listed on the Job Management tab and then runs the job. If no jobs have been created, the export cannot be saved. If you do not have Job Management permissions, this option is not displayed.

- **Run Now** – runs the export job.

5. If you opt to Save or Save & Run Now, you can select the job where you want to save the export to. A success or fail message is displayed. If no job has been created, you can either create a job (if you have permissions to work on Table Job Management) or you can Run Now.

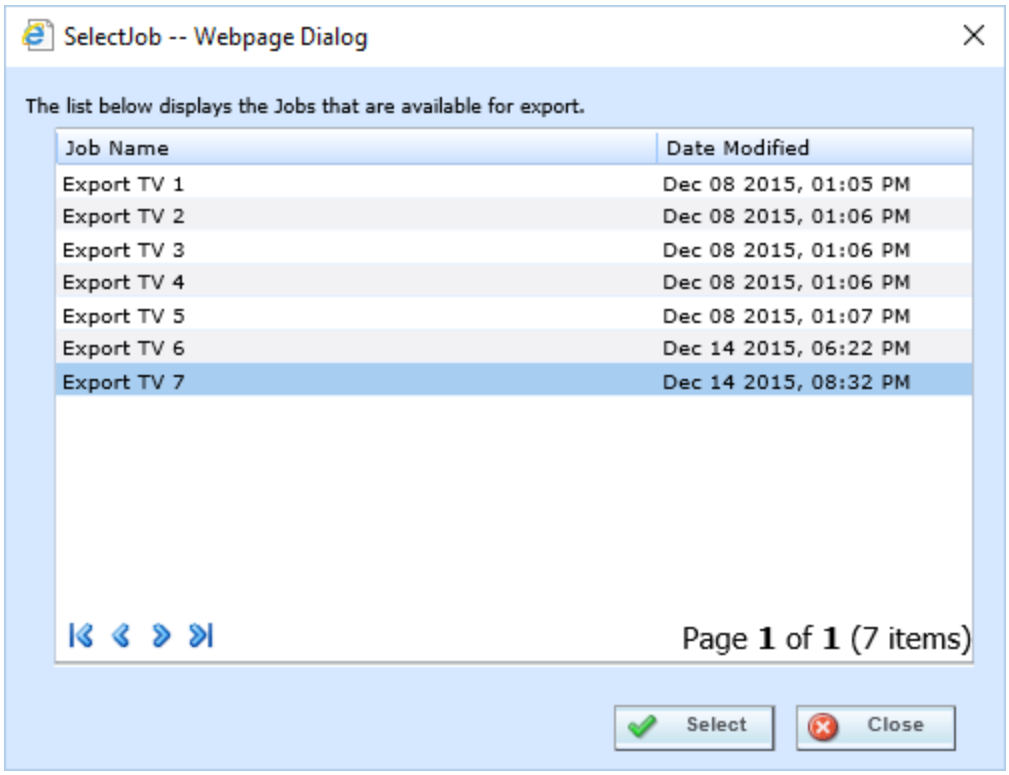


Figure 214 Selecting a Job

6. After the exports have been saved, users with permissions can view, edit and start the job on the Table Job Management screen. If you choose to Run Now, no job needs to be started. The Excel file will be listed on the library.

7. Once the job has been run, the export is displayed the table export page of the Library. This is open to all users.

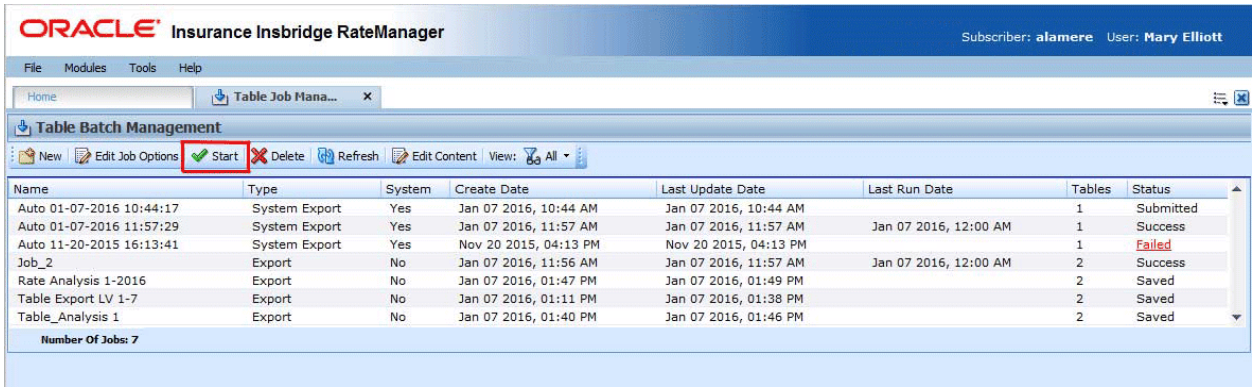


Figure 215 Jobs Listed on Table Job Management

Export Data File

On The Export Data File screen, the Destination Workbook name is displayed. The default value is the table variable name. You can edit the name but not the file extension. The default file type extension is .xlsx. If you name a workbook RateFeeGlobal.xls that will be the workbook name: RateFeeGlobal.xls.xlsx. The file will not be saved as an Excel 93 – 2007 file type.

The Worksheet Name is displayed. The default value is also the table variable name. You can edit the name. When creating a bulk table export job, each table is returned as a single workbook. One job can return multiple workbooks.

There is an option to save to an existing file. This saves the job to an existing file on the server. Click the File button to open the saved files. Using an existing Workbook will overwrite all content.

Overwriting Existing Files with Export to MS Excel

Files are overwritten when the table job export is run again and the previous file is still there. A warning message is displayed when you select to Save, Save & Run Now or Run Now.

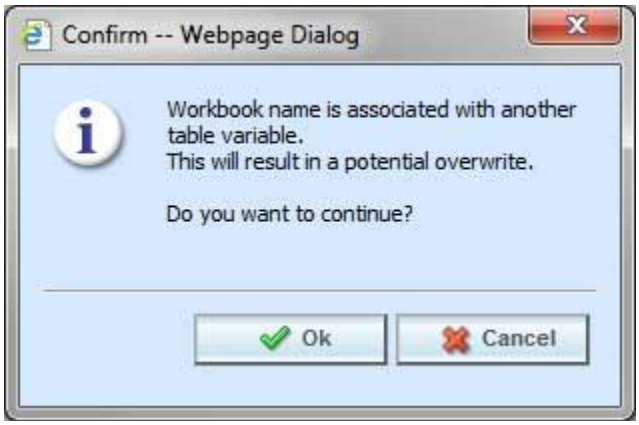


Figure 216 Overwriting a Job

If the file exists, a warning message is displayed to let you know that the file will be overwritten. Click OK to overwrite the file or Cancel to return to the export screen.

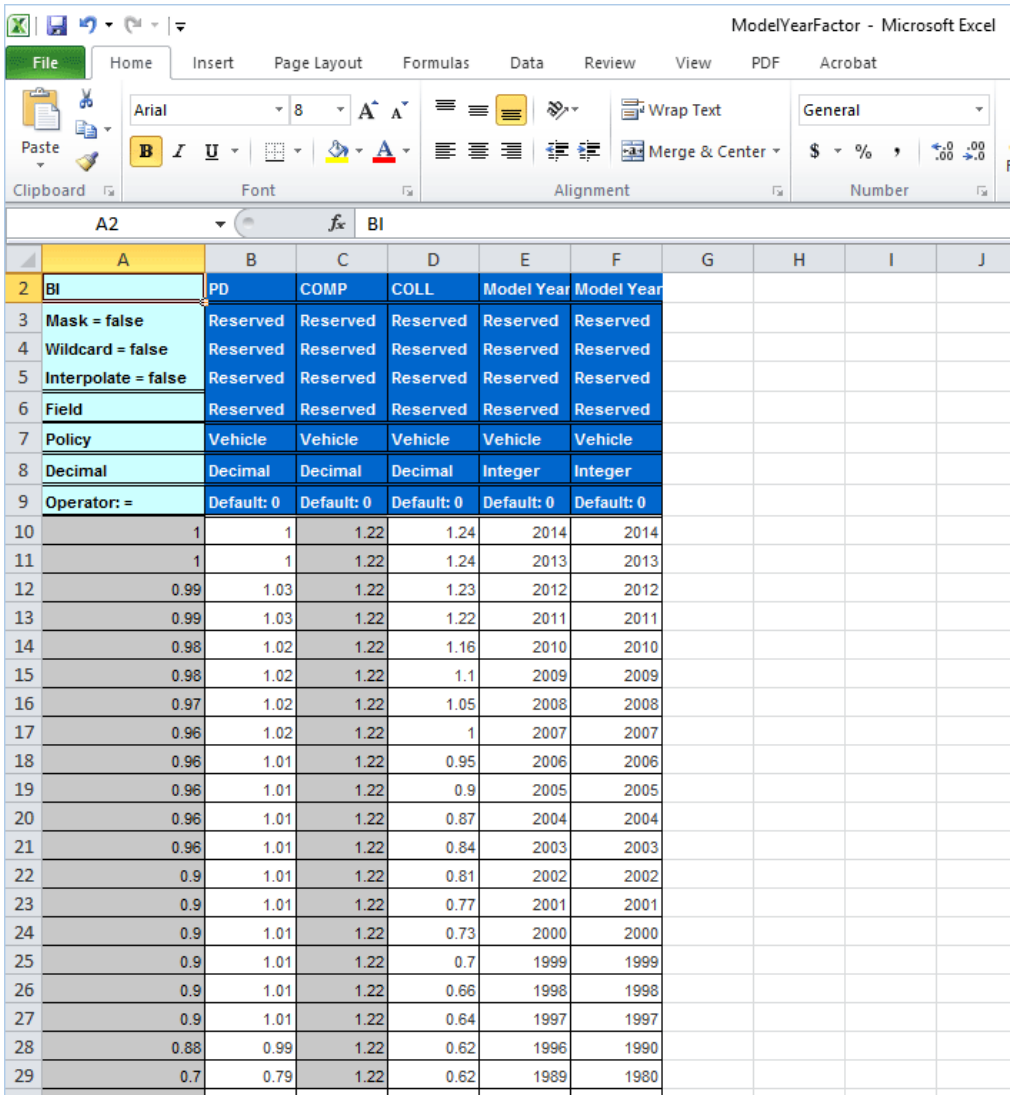
If you want a second copy of this table variable to be exported, change the Workbook Destination name. This creates a new file using the updated workbook name.

If the first file is no longer needed, delete the job that created the file from the Table Job Management screen. This removes the file and you can export the table.

For more information, please see the Insbridge RateManager Admin User Guide under Table Job Management.

EDITING AN MS EXCEL FILE

Excel files that are immediately opened can be viewed and edited but are not automatically saved. If you want to save the file, you must save it in Excel. The downloaded file may need to have editing enabled.



	A	B	C	D	E	F	G	H	I	J
2	BI	PD	COMP	COLL	Model Year	Model Year				
3	Mask = false	Reserved	Reserved	Reserved	Reserved	Reserved				
4	Wildcard = false	Reserved	Reserved	Reserved	Reserved	Reserved				
5	Interpolate = false	Reserved	Reserved	Reserved	Reserved	Reserved				
6	Field	Reserved	Reserved	Reserved	Reserved	Reserved				
7	Policy	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle				
8	Decimal	Decimal	Decimal	Decimal	Integer	Integer				
9	Operator: =	Default: 0	Default: 0	Default: 0	Default: 0	Default: 0				
10	1	1	1.22	1.24	2014	2014				
11	1	1	1.22	1.24	2013	2013				
12	0.99	1.03	1.22	1.23	2012	2012				
13	0.99	1.03	1.22	1.22	2011	2011				
14	0.98	1.02	1.22	1.16	2010	2010				
15	0.98	1.02	1.22	1.1	2009	2009				
16	0.97	1.02	1.22	1.05	2008	2008				
17	0.96	1.02	1.22	1	2007	2007				
18	0.96	1.01	1.22	0.95	2006	2006				
19	0.96	1.01	1.22	0.9	2005	2005				
20	0.96	1.01	1.22	0.87	2004	2004				
21	0.96	1.01	1.22	0.84	2003	2003				
22	0.9	1.01	1.22	0.81	2002	2002				
23	0.9	1.01	1.22	0.77	2001	2001				
24	0.9	1.01	1.22	0.73	2000	2000				
25	0.9	1.01	1.22	0.7	1999	1999				
26	0.9	1.01	1.22	0.66	1998	1998				
27	0.9	1.01	1.22	0.64	1997	1997				
28	0.88	0.99	1.22	0.62	1996	1990				
29	0.7	0.79	1.22	0.62	1989	1980				

Figure 217 Exported Table in Excel

Table Composition

When the file is opened, the file can be edited as needed. The light blue columns on the left are the criteria and the dark blue columns are the table variable and linked table variables, if applicable.

The rows under the criteria and table variables are the current values.

Criteria Definition

Criteria definitions start in Column A, Row 2 (A2). All criteria are listed. The end of the criteria list depends on the number of criteria that are defined in the table. The maximum number of criteria allowed in a table variable is 50.

Row	Contents	Notes
2	TableCriteria Name	The name of the variable that defines the criteria values
3	Mask =	Masking applies - 'true' or 'false'
4	Wildcard =	Wildcarding applies - 'true' or 'false'
4	Interpolate =	Interpolation applies - 'true' or 'false'
6	VarType	The type of variable that defines the criteria values • Fields • CalcVar • TableVar
7	Category	The category of the variable that defines the criteria values
8	DataType	The data type enforced for the values stored • Integer • String • Decimal • Date
9	Operator	The operator used to determine whether the incoming data match is 'true' or 'false' • [=] equals • [<>] not equals • [>] greater than • [<] less than • [>=] greater than or equal to • [<=] less than or equal

Figure 218 Criteria Definitions for Exported Table Variables

True = the option is on the table variable criteria

False = the option is not on the table variable criteria

Table Identifiers & Results (Table) Definition

The table variable definition start column depends on the number of criteria defined in the table. There is a maximum number of criteria is 50, column AY. The last column used depends on the number of linked table variables defined.

Row	Contents	Notes
2	TableVar	The name of the table variable (or linked table variable) that defines the result values in this column
3	TableID1	Reserved
4	TableID2	Reserved
5	TableID3	Reserved
6	TableID4	Reserved
7	Category	The category of the variable that defines the result and table values

Row	Contents	Notes
8	DataType	The data type enforced for the values stored • Integer • String • Decimal • Date
9	Default: n	Then default value that is used at runtime if no matching criteria row is found

Figure 219 Table Identifiers and Results Definitions

Table Data

Table data begins at row 10. The number of rows used depends on the amount of data in the table. All data is current for the revision selected to be exported.

Row	Column	Notes
10	A	The beginning of the table data values

Figure 220 Table Data Definitions

All exported tables use a system-defined format with row 1 being used for system defined properties. The export format cannot be modified and exported tables cannot be imported back in to RateManager at this time.

Table data that has been exported can be edited in Excel and re-imported back in to RateManager. When re-importing, select the Insbridge format. This starts the data import from row 10.

EDITING TABLE DATA

The Edit Data option on the Edit Variable tab opens the Edit Data screen. Editing on cell or rows can be done on the Edit Data screen.

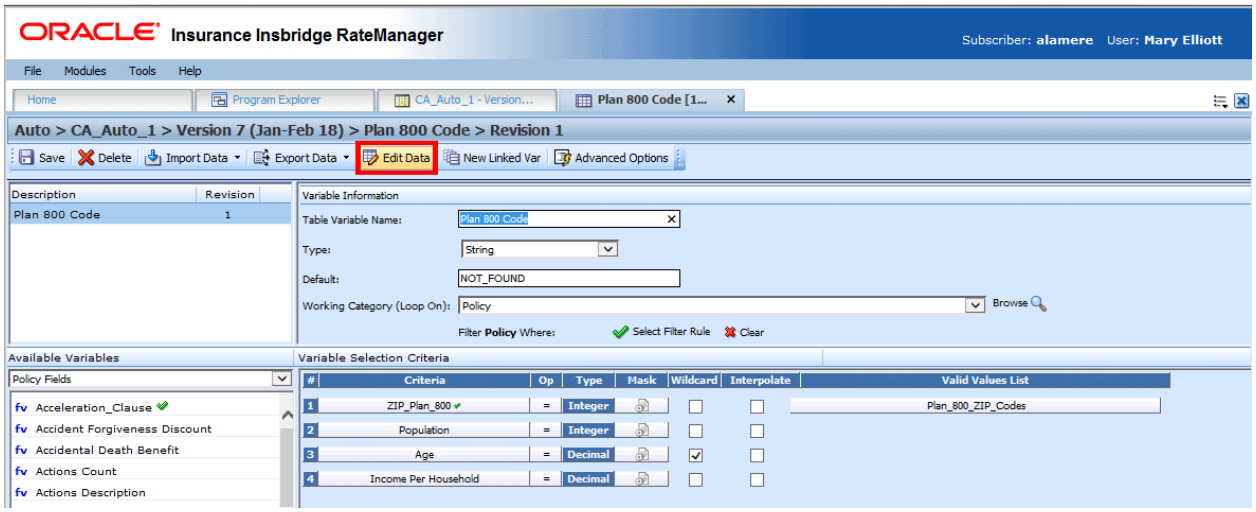


Figure 221 Editing Table Data

NOTE: *If you are on the **Edit Variable** tab and you have changed the criteria information or added/deleted linked variables, remember to save your variable before editing the data.*

Edit Data

Table data can be edited directly in the table. Data within an individual cell can be added, deleted or updated. Single or multiple rows can be added or deleted or exported. Data can be sorted and filtered.

For variables that are locked, such as template generated variables, data can be viewed in the data preview area of the Program Details tab but changes are not allowed. A new revision can be created that will be open for editing.

If a lot of changes are needed on a table, you may want to export the table, edit it, and then re-import the table. If you are using the table in a SoftData call, be aware that database row numbers may be updated.

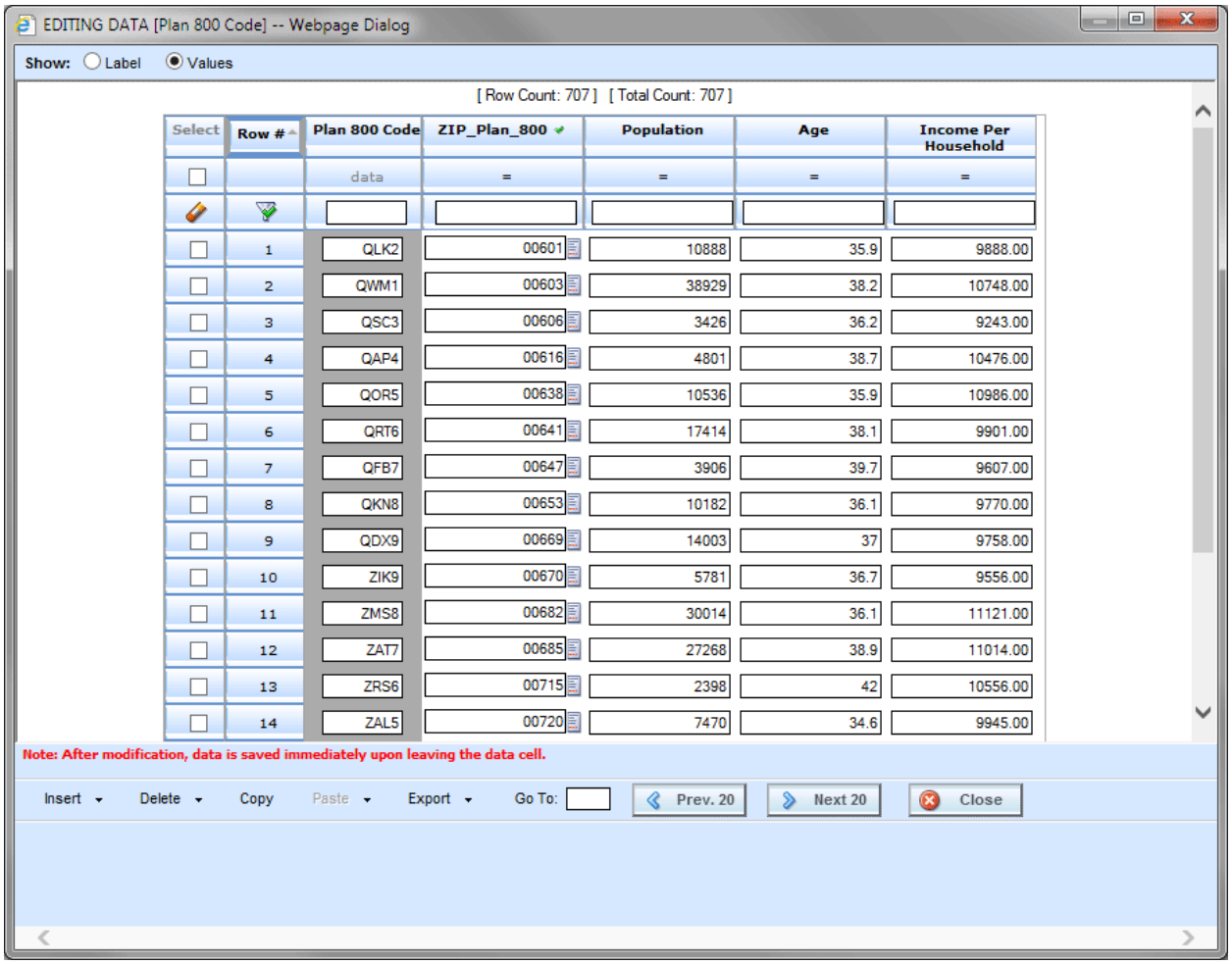


Figure 222 Navigating Edit Data Screen

The Edit Table Data screen displays the current table data. Editing and navigation options are at the top and bottom of the screen. If a list of values has been applied, the options to change the view are at the top of the screen. This screen must be closed before viewing or working on any other tab in RateManager.

Row Options

A row count for the entire table is listed at the top of the screen. Row numbers at the front of the row reflect the database row. The row count is updated as rows are added or removed. The database row number does not

change when rows are removed. When rows are added, a database row number is assigned to the new row. Rows following the new row may have the database row number updated also. All database row numbers are updated when the entire table is re-imported. Database row numbers may be used in other functions in RateManager for example, in SoftData calls.

The button options at the bottom of the screen are for rows only.

- Insert – Inserts a row either above or below the selected row.
- Delete – Removes the select row, or selected rows, or all rows on the page.
- Copy – Copies the selected row, or selected rows.
- Paste – Pastes copied row(s) either above or below the selected row. If a row is not selected, the rows are pasted at the end of the table. Paste is only displayed when a row has been copied.
- Export – Exports filtered rows in either tab delimited or MS Excel format.

Right Click Menu

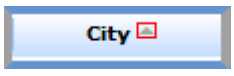
Cells in the data table have a right click menu where you can select all contents in the cell, paste in value, or delete values. To use a right click menu option, select the cell where you want to edit data and right click. The changes are immediately saved when you exit the cell.

Paging

If the table contains more than twenty rows, use the **Prev.20** and the **Next 20** buttons to navigate through the table. You also can go directly to a row by entering the table row number in the **GO TO box** and clicking anywhere outside the box.

Icons and Options

The Edit Data screen contains icons and options to assist with editing table data.

Icon	What it Indicates
	Clear – Clears the filter criteria.
	Filter – Filters the entire table. The rows returned are only the rows that meet the filter criteria.
	Filter Removed – Removes the filter criteria.
	**Sort** – Sorts table data by the selected column in ascending or descending order. The column name is highlighted and the ascending or descending arrow is displayed.
	Show – When working with field variables that use a list of values, users can select the view for the valid values column. If there are no valid values, the view does not change.

Editing Data in a Cell

1. Navigate to the **Edit Variable** tab for the table variable where you want to edit data.

2. Click **Edit Data** to open the edit screen.

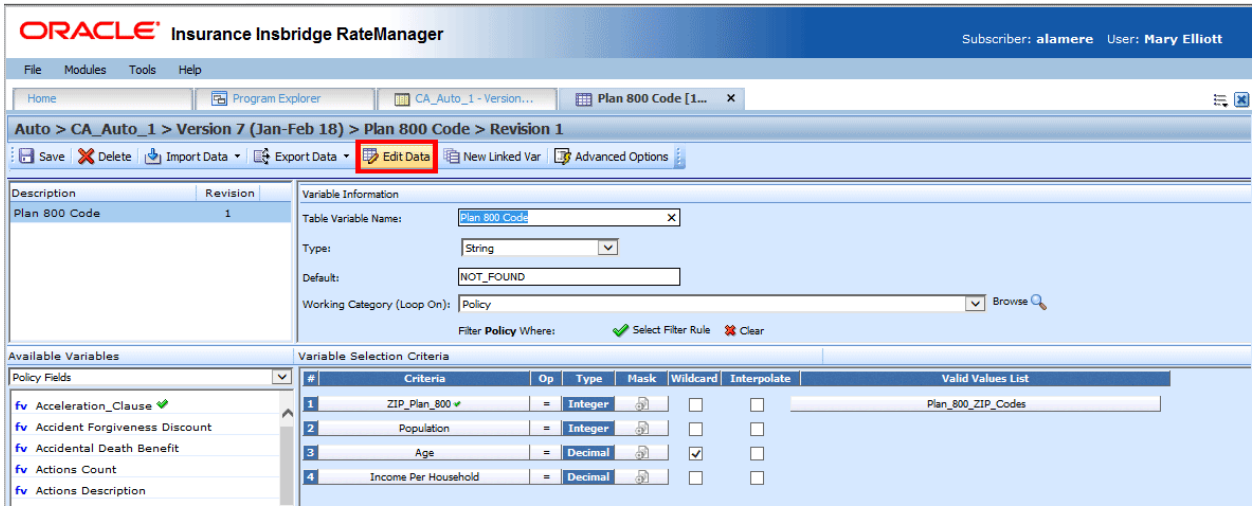


Figure 223 Editing Table Data

3. Locate the cell to be edited.

4. Click on the cell to be edited and enter the updated data. Double clicking opens a separate window where edits can be done. If there are additional cells to be edited, you can click on the next cell or tab to it. Edits are saved as soon as you leave the cell.

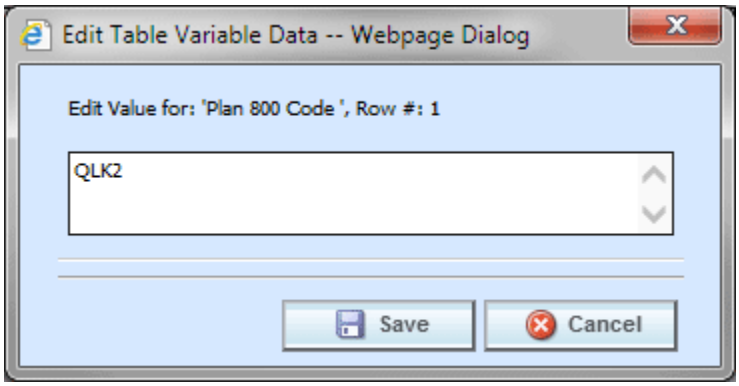


Figure 224 Edit Table Variable Data

NOTE: *When you make a change to a cell in the data table, it is immediately updated in the database. Changes to the data table cannot be undone by clicking **Cancel** on the Edit Data screen. You must reenter the data if you inadvertently change a value.*

5. When you are finished editing the data, click **Close** to return the **Edit Variable** tab. If you changed any of the data, the time stamp on the **Program Details** tab is updated.

Editing Rows

Rows can be inserted, deleted, copied, pasted or exported. The Insert, Delete, Copy, Paste, and Export options are on the lower left portion of the screen, next to the Previous 20 button.

1. Navigate to the **Edit Variable** tab for the table variable where you want to edit data.

2. Click **Edit Data** to open the edit screen.

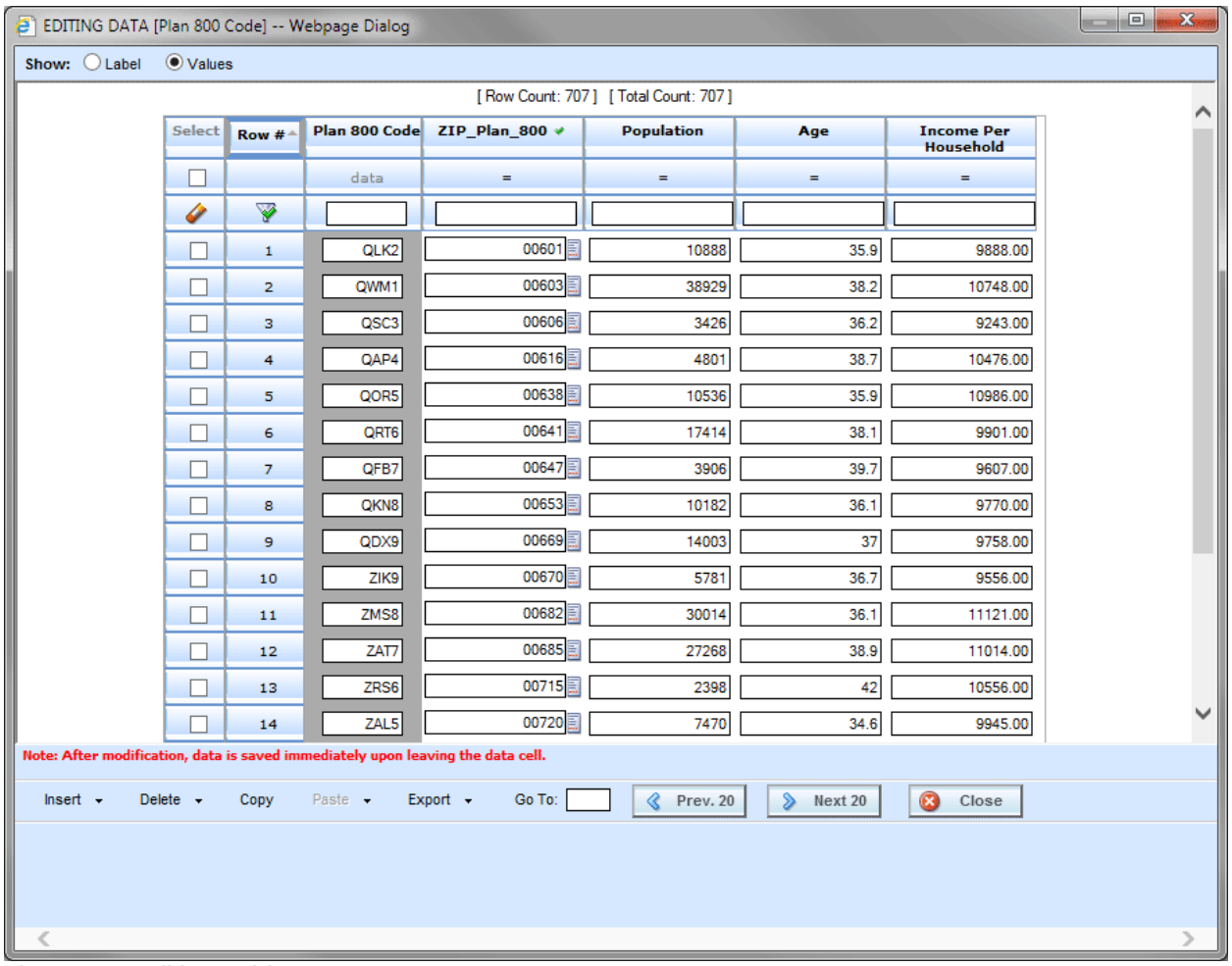


Figure 225 Editing Table Data

3. Locate the row to be edited. If the table contains more than twenty rows, use the **Prev.20** and the **Next 20** buttons to navigate through the table. You also can go directly to a row by entering the row number in the **GO TO box** and clicking anywhere outside the box.

4. When you are on the row you want, you can insert, delete or copy and paste.

Inserting Rows

Rows can be inserted any place in the table. If a specific row is not chosen, inserted rows are added to the end of the table. You can only insert one row at a time.

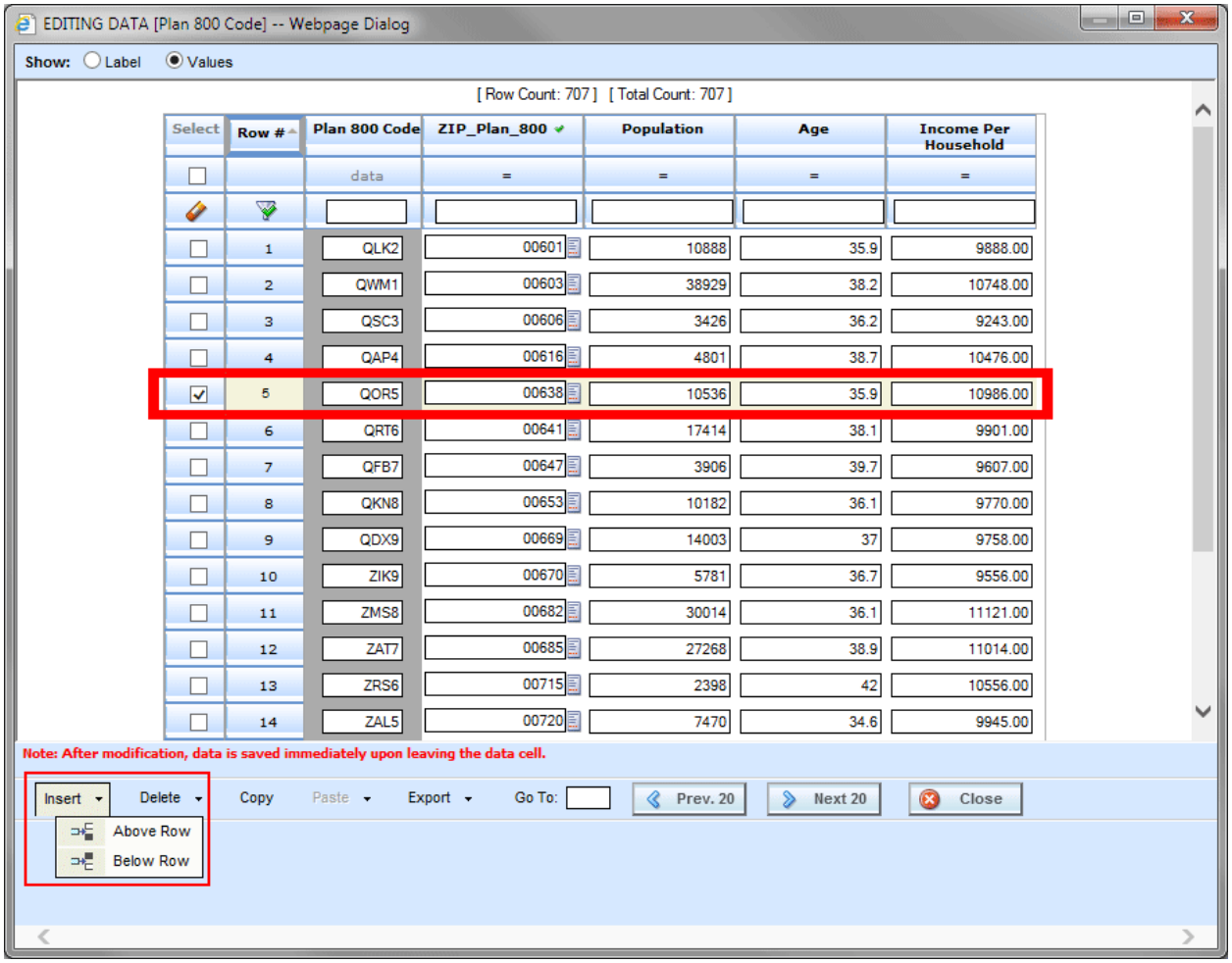


Figure 226 Inserting Rows on Table Variable Data

1. To insert a new row in to a specific place in the table, select a cell in row above or the row below where you want the new row to be placed. The row number will be highlighted and the selected cell will have a bold outline.

2. Click **Insert** and then select either **Above Row** or **Below Row** to insert the new row either above or below the selected row. If no cell is selected, the new row will be added to the end of the table.

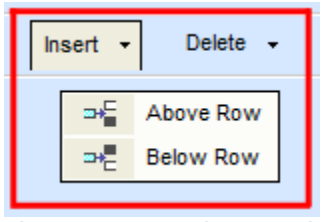


Figure 227 Inserting Row in Table

Deleting Rows

A single row or multiple rows can be deleted from the table. There is a warning message prior to the row or rows being deleted. Deletions are per page. You cannot select multiple rows across multiple pages and delete. Only the rows on the page you are currently on can be deleted. Make sure this is the action you want to take. Deletions cannot be undone.

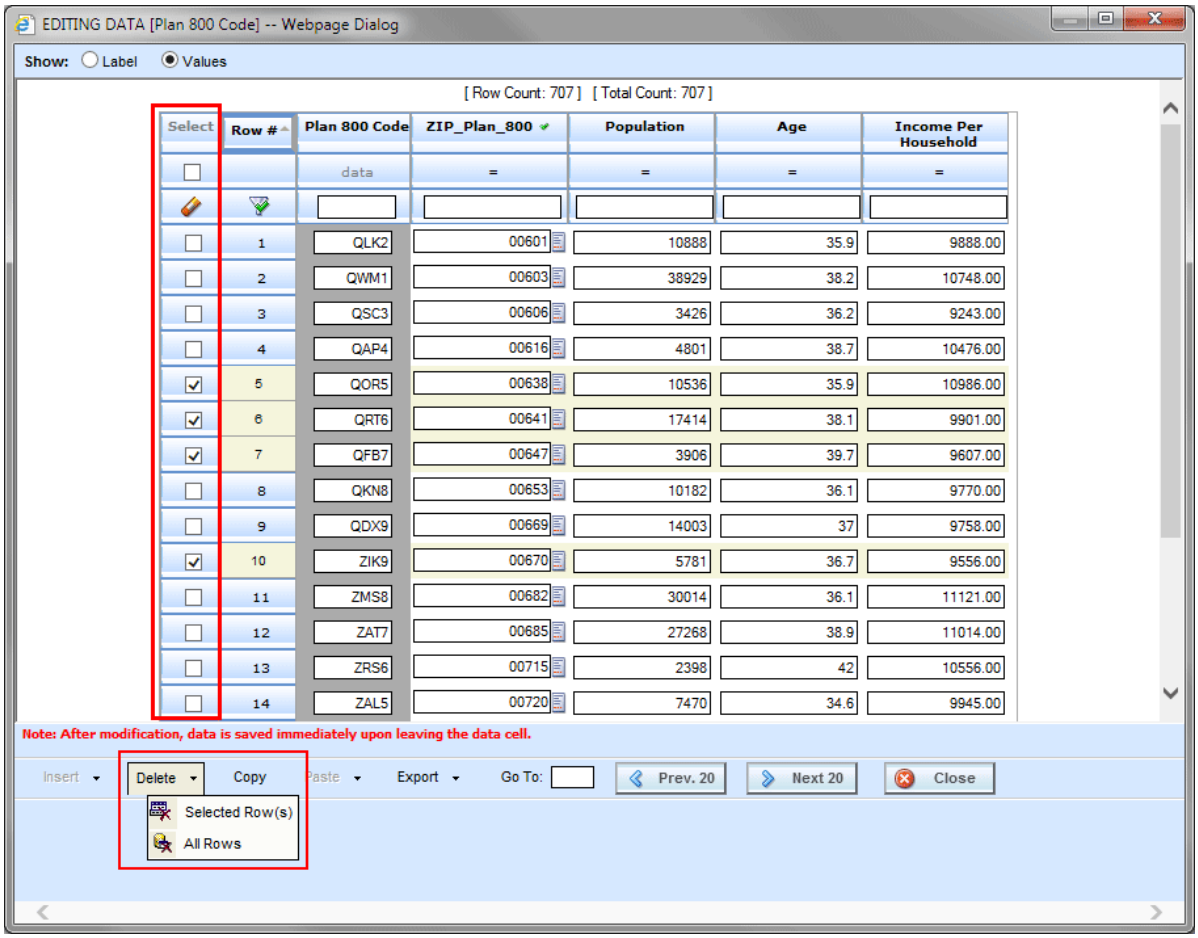


Figure 228 Deleting Rows from Table Data

1. To delete a single row from the table, select a cell in the row you want to delete. Make sure the cell is highlighted with a bold outline. Click **Delete** and **Selected Row(s)**. You will be asked to confirm deleting the row.

2. To delete more than one row, check the boxes in the Select column of the rows you want to delete. Verify your selections. Click **Delete** and **Selected Row(s)**. You will be asked to confirm deleting all rows.

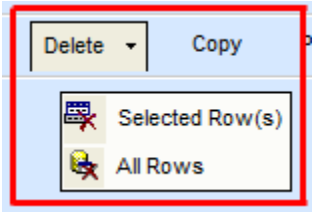


Figure 229 Deleting All Rows in Table

3. To delete all rows from the table, click **Delete** and **All Rows**. You will be asked to confirm deleting all rows.

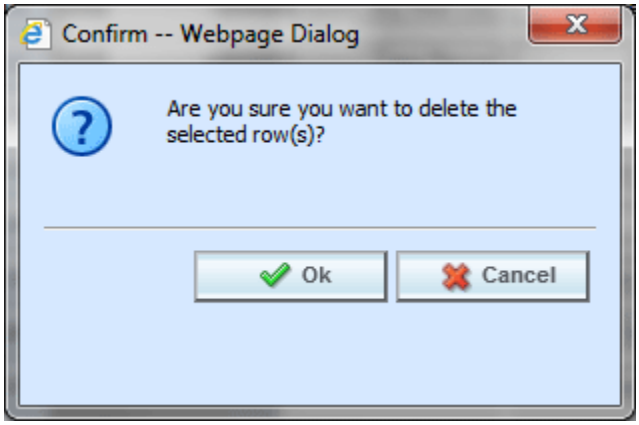


Figure 230 Confirming All Rows Delete

4. Select **Ok** to delete selected rows or all rows and return to the Edit Data screen or **Cancel** to return to the Edit Data screen without deleting any rows.

Copy Rows

Any row can be copied and pasted anywhere in the table. If no row has been selected, the row will be pasted at the end of the table.

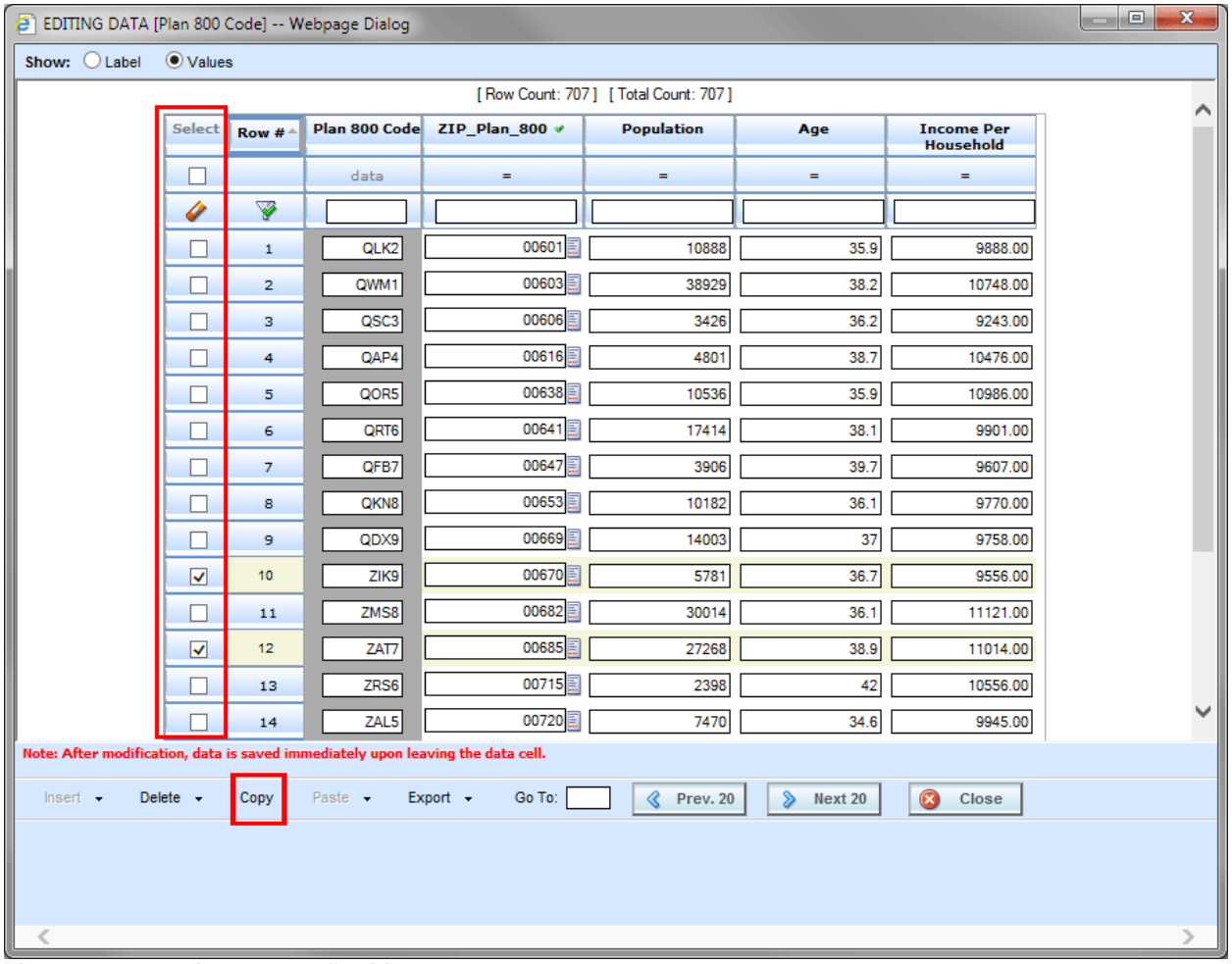


Figure 231 Copying Rows of Table Data

1. To copy one or more rows, check the boxes in the Select column of the row(s) you want to copy. Verify your selections. Click **Copy**.

2. The screen updates and the copy selection(s) is highlighted. The paste options now are available.

NOTE: *If this is not the action you want to take, click the Esc key on your keyboard, this cancels the copy.*

Pasting Rows

Copied rows can be pasted any place in the table. If a specific row is not chosen, copied rows are added to the end of the table.

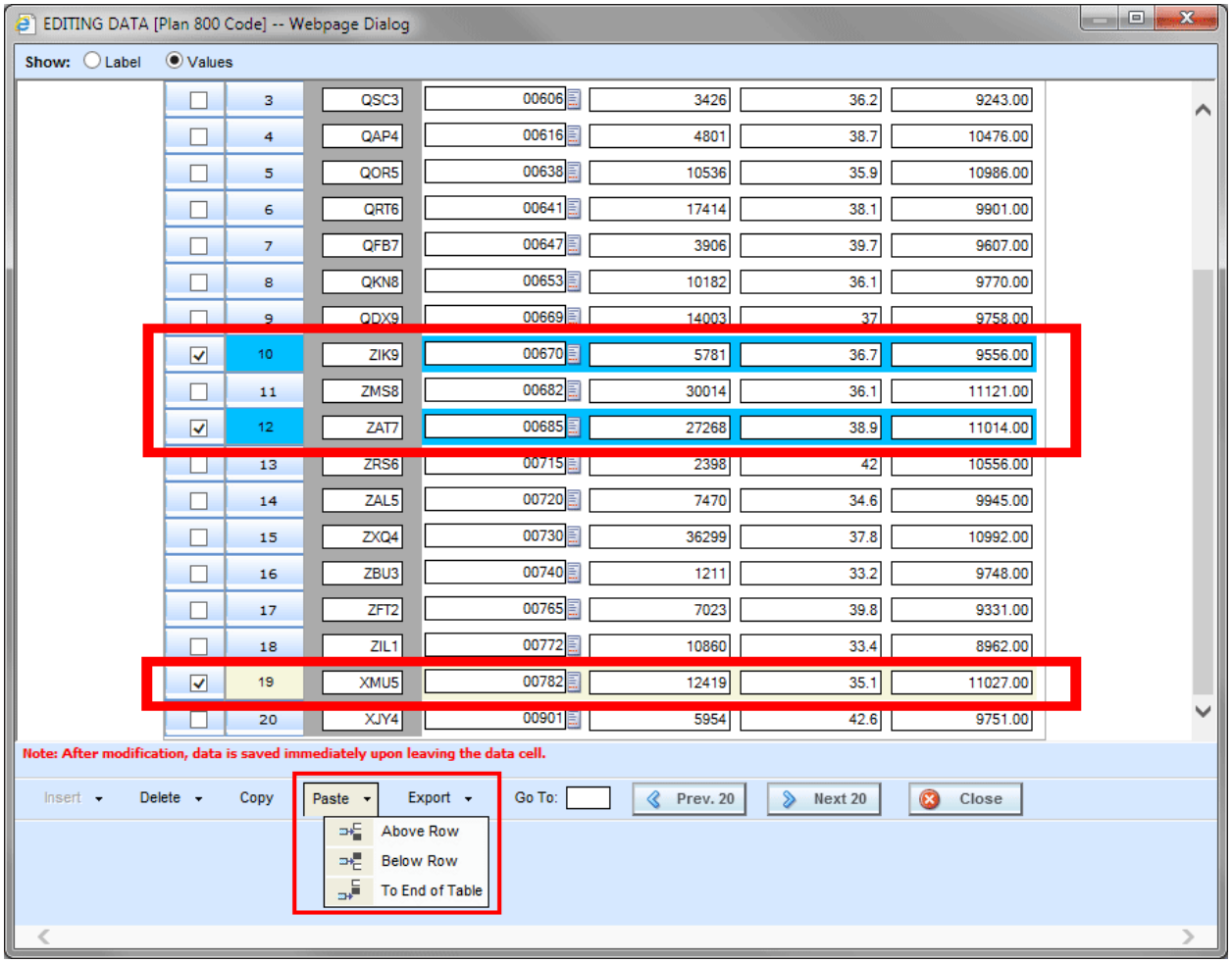


Figure 232 Pasting Rows of Data in a Table

1. When a copy of a row is made, the row that has been copied is highlighted. In the example above, the row is bright blue.

2. To paste a copied row in a specific place in the table, select the row above or the row below where you want the copied row to be placed. The row number will be highlighted and the Select field will have a check.

3. To past a copied row at the end of the table, you do not need to select any row.

4. Click **Paste** and then select **Above Row**, **Below Row**, or **To End of Table** to paste the copied row. If no row is selected, the copied row will be added to the end of the table.

Filtered Rows

Table data can be filtered to make it easier to locate or edit data within RateManager. Filters can be added to any column in the table and the data returned is only the data that matches the filter criteria. The match must be exact. The filter is done on the entire table.

Data can be filtered by one or more columns. Additional criteria can be entered to further define the data returned. Filters can be removed to expand the data returned. If a filter is in place, any data that does not match to filter is not presented. The filter happens as so as you leave the field.

Filtered data stays in place until the filter is removed or updated. Filtered data can be sorted or exported. The filter is for editing purposes only and does not remove data from the database or table.

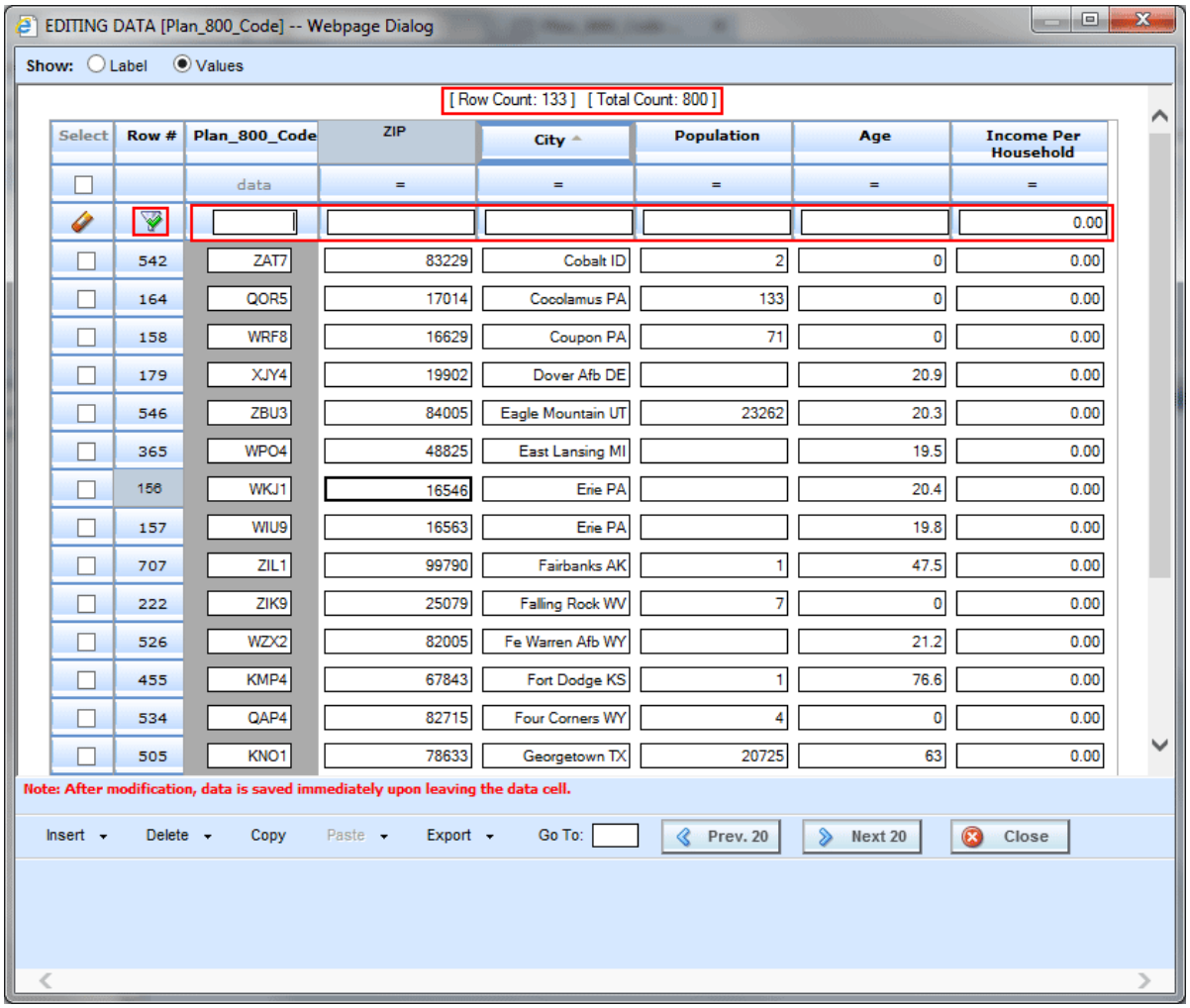


Figure 233 Filtered Table Data

1. To filter all the data in the table, enter filter criteria the field at the top of the column where filtering is to be done. If you exit the field, the filtering is done when you leave the field. But filtering also can be done by entering a value and then clicking the filter icon.

2. The data returned is the data that meets the filter criteria. Additional criteria can be entered to further define the data returned.

Suspending a Filter

To suspend filtering, click the filter icon a second time. The icon is displayed with a red mark and the filter values are grayed out. This suspends all filtering. If you want to resume filtering or edit the filter criteria, click the filter icon again.

When filtering is suspended, the entire data table is returned and the row count is updated.

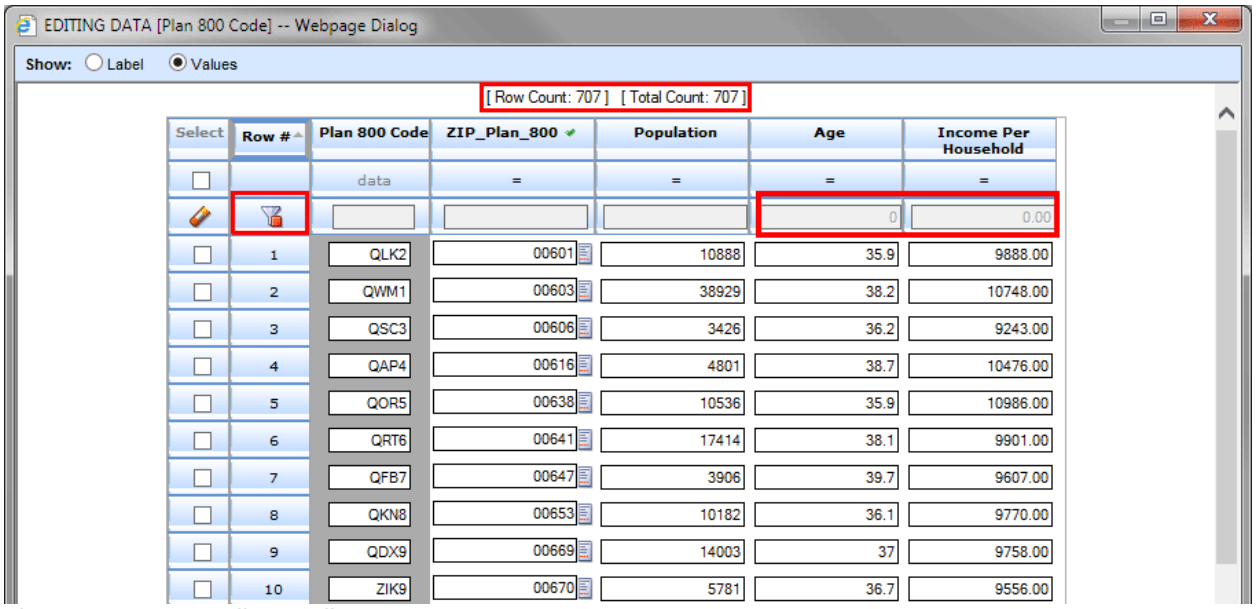


Figure 234 Suspending a Filter

Removing a Filter

To remove a filter, click the eraser icon . All filter criteria are removed from the top field and the entire data table is returned.

Filtering Table Data with a List of Values

When working with table data that contains a list of values, filtering is done on the value, not on the label. To make sure the desired results are displayed, select the values view in the Show option. A column with a list of values has a green checkmark and a list of values icon next to the data field.

Sorting Rows

Table data can be sorted to make it easier to locate or edit data within RateManager. Data can be sorted by any single column. The sort is applied to the entire table and remains in place when paging through the data or applying a filter.

The default sorting row is the row number. If sorting by another column is no longer needed, click the row number column to return to the default.

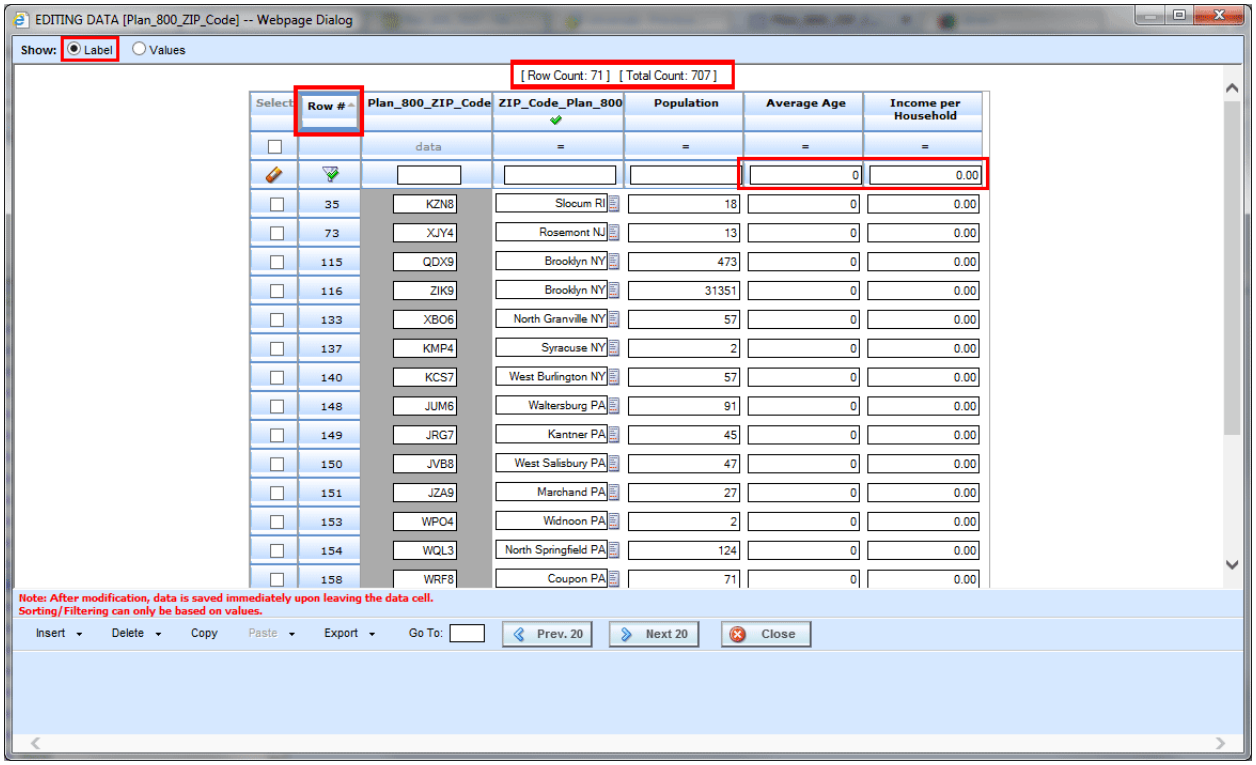


Figure 235 Sorted and Filtered Table Data

- To sort by a specific column, click the column header. The header is highlighted and the sort arrow is displayed.
- A downwards arrow indicated descending order, and up arrow is ascending order. The default to ascending from lowest to highest. To switch to descending, click the header a second time.
- Sorting can be done on data that is filtered or uses a list of values.
- Sorted data can be in ascending or descending order. The sorting values may differ from MS SQL Server where numeric values are before special characters. Sorting values are:

Ascending	Descending
Blank	Alphabetical
Special Characters	Numeric
Numeric	Special Characters
Alphabetical	Blank

Exporting Rows

The Export option at the bottom of the edit data screen allows for filtered rows of data to be exported in tab delimited or MS Excel format. Exporting smaller sections of data can save time and reduce the risk of an accidental edits to data that did not change. Data can be filtered by one or more columns. The data returned is the data that meets the filter criteria. When the required subset of data is complete, the data can be exported in tab delimited format or MS Excel format.

For example, if there is a table with 100,000 rows and changes need to be made to 50 rows with common data values, table data can be filtered down and the filtered rows can be exported in MS Excel format. After any edits, the data can be re-imported and appended to the table.

The number of rows to be exported from the edit table data screen is listed at the top of the screen. All data that meets the filters is exported. Exports are not done per page.

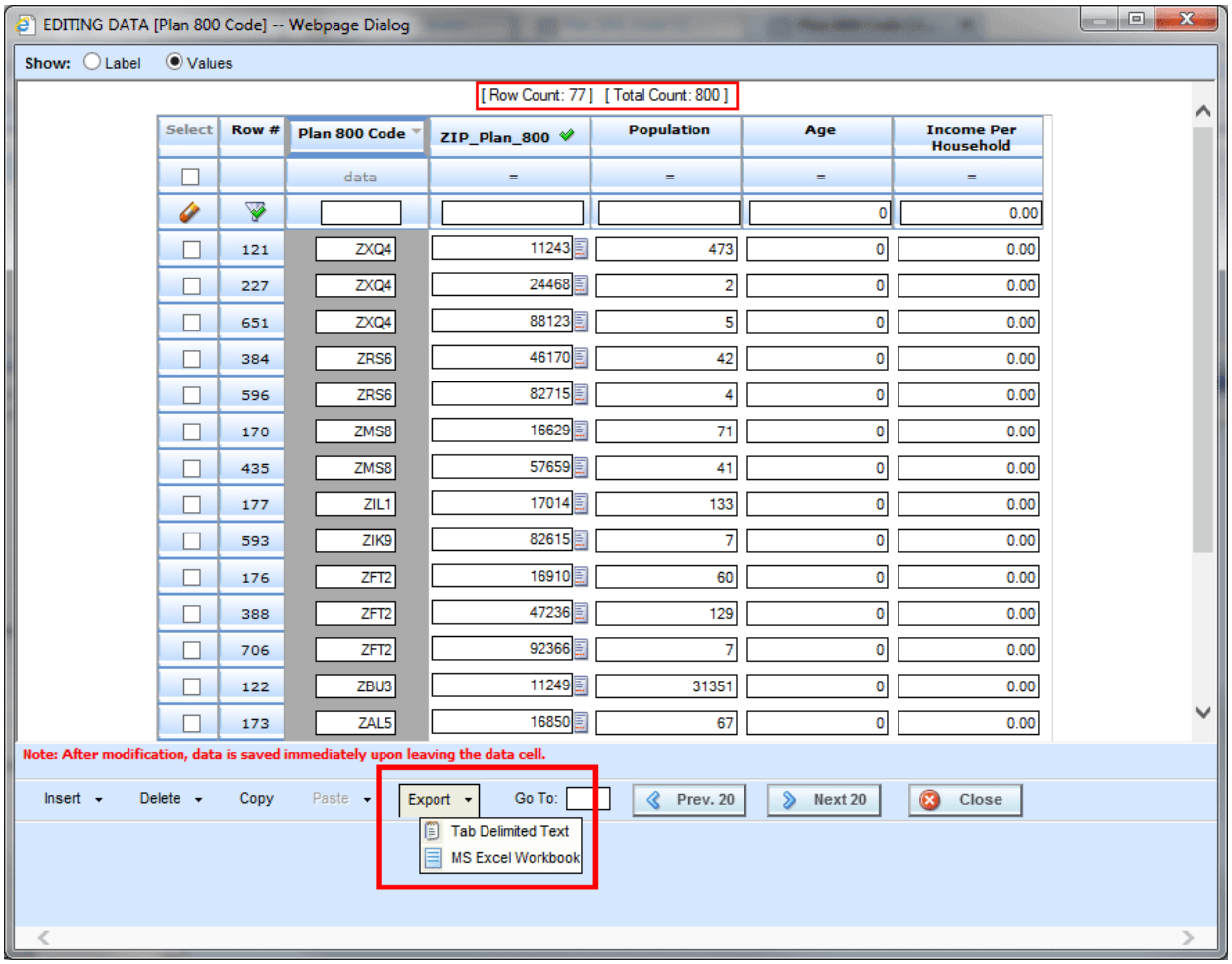


Figure 236 Filtered Data for Export

1. On the Edit Data screen, enter filter criteria for the rows or rows where you want to filter data.

2. Click the filter icon. The screen refreshes with the data that matches the filter. The number of rows returned is listed at the top of the screen.

3. When the subset of data is complete, select the type of export needed; tab delimited or MS Excel.

Tab delimited reports are presented immediately. MS Excel reports can be saved to a table job or run immediately. Please see Export Data to MS Excel Workbook for more information.

To allow for exports to MS Excel, the same setup procedures for table data exports must be in place.

Data exported in tab delimited format cannot be appended to the table.

Working with a List of Values

When working with a table that contains a list of values, sorting is done on the value, not on the label. If you sort on a list of values with the labels displayed, the sort order may not look correct. Change the view to Value to see the correct sorting. To make sure the desired results are displayed, select the values view in the Show option.

A column with a list of values has a green checkmark and a list of values icon next to the data field.

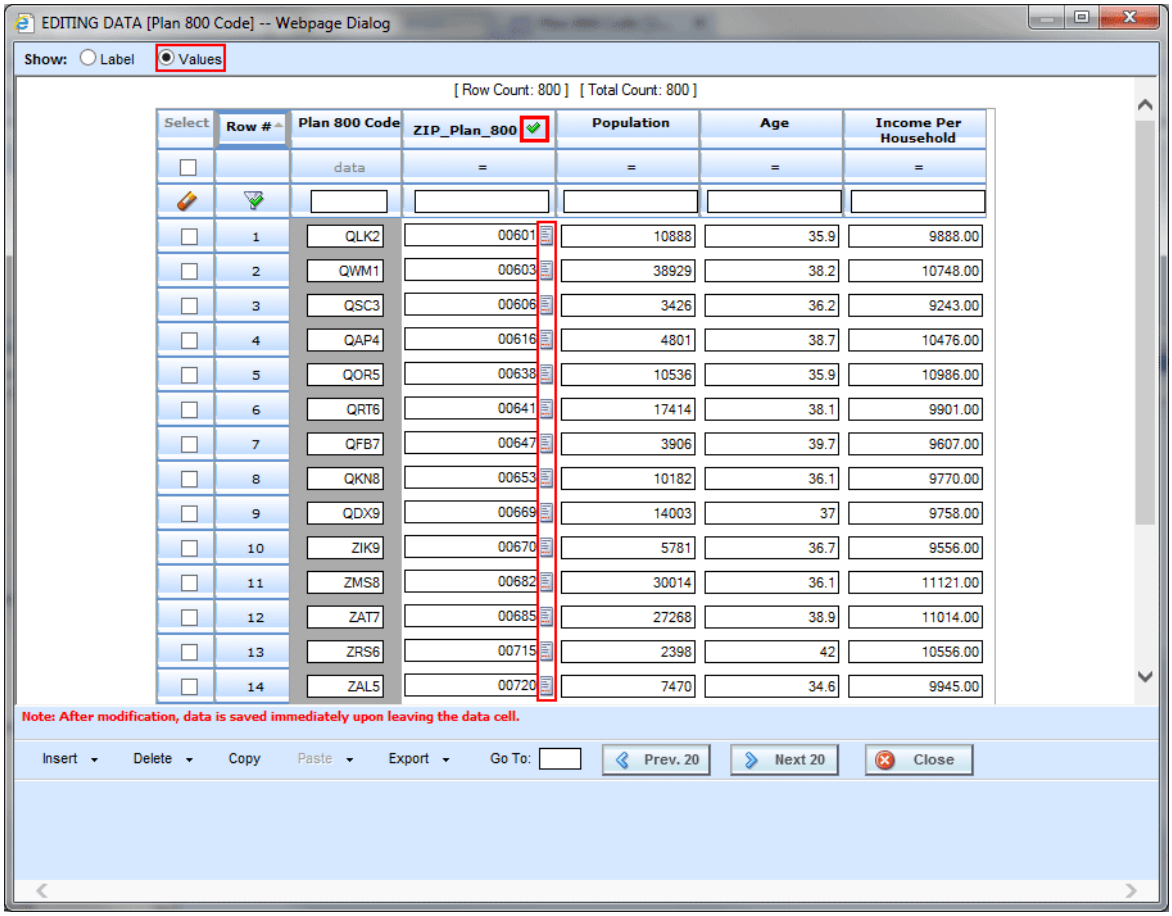


Figure 237 Table Data with a List of Values

When a list of values icon is clicked, the valid values for the field are listed. The first 25 values are listed regardless of where you clicked the icon. If you click the list of values icon on row 600, the first 25 values in the table are displayed. If you need a value beyond the first 25, you can search for it.

Searches are done on the label. To do a search, enter your criteria in the search field and matches will populate the listing. Up to 25 labels that match the search criteria are displayed.

Cells on the table data can be updated by selecting a label/value and clicking Use Value. Changes are immediate and cannot be undone.

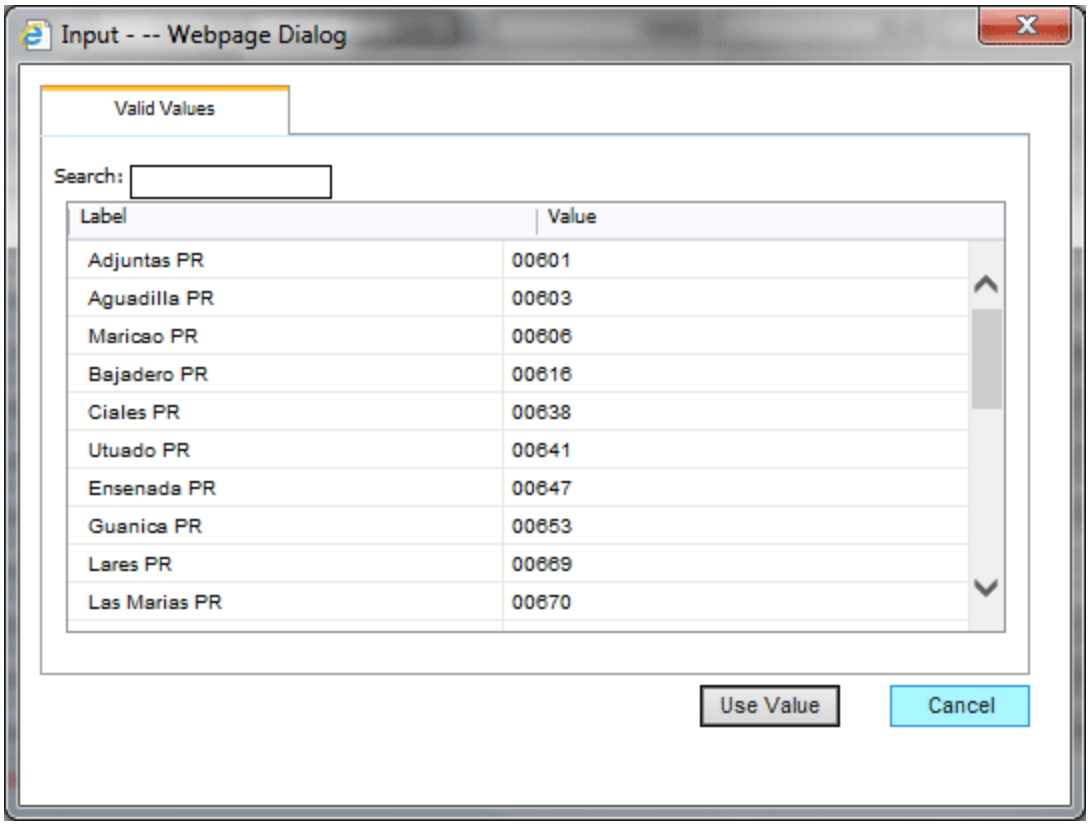


Figure 238 Valid Values for Table Data

Changing the View

When a field with a list of values is included in a table variable, you can choose to view the field by the value or the label. The default view is the value.

Changing the Show option changes the display for the currently selected screen, unless a filter is in place. When a filter is in place, the show option will stay for the rows returned.

When the Label option is selected, the label is displayed in the cell.

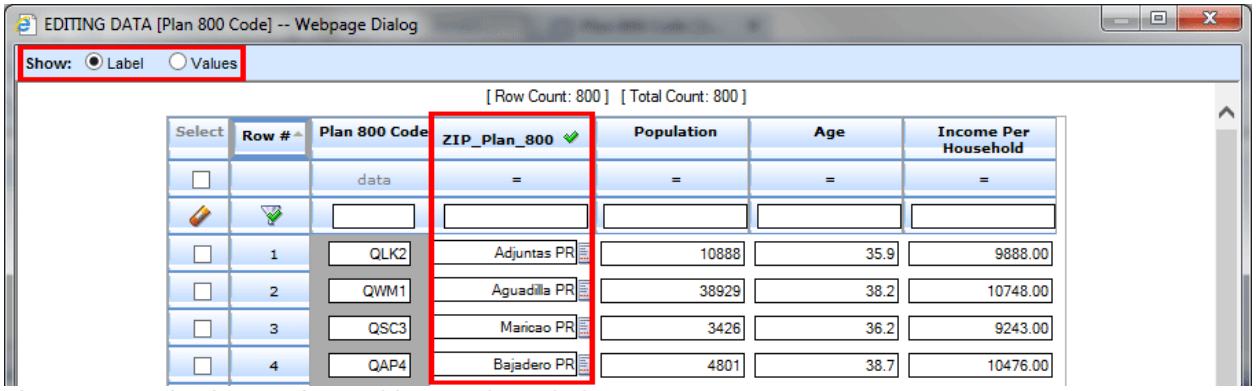


Figure 239 Selecting to View Table Date by Label

DATA SOURCES

An Insbridge data source is a special table type that defines table criteria columns without using existing RateManager variables. Data sources are similar to table variables in that they lookup data based on one or more criteria and depending on the values of those criteria, a single row is selected and returned as the result of the table lookup.

Data Sources differ from table variables in that the lookup criteria do not come from a static set of existing variable elements (Fields, Calculated Variables, Table Variables, and Result Variables). The criteria columns of a data source can be defined to accept a wide variety of criteria. Data sources also allow for criteria to be grouped on the table. Another difference is that data sources are not used in any calculated variable or algorithm but are used exclusively in SoftData calls.

Data sources are viewed and managed on the table variable tab and are represented by a unique icon.

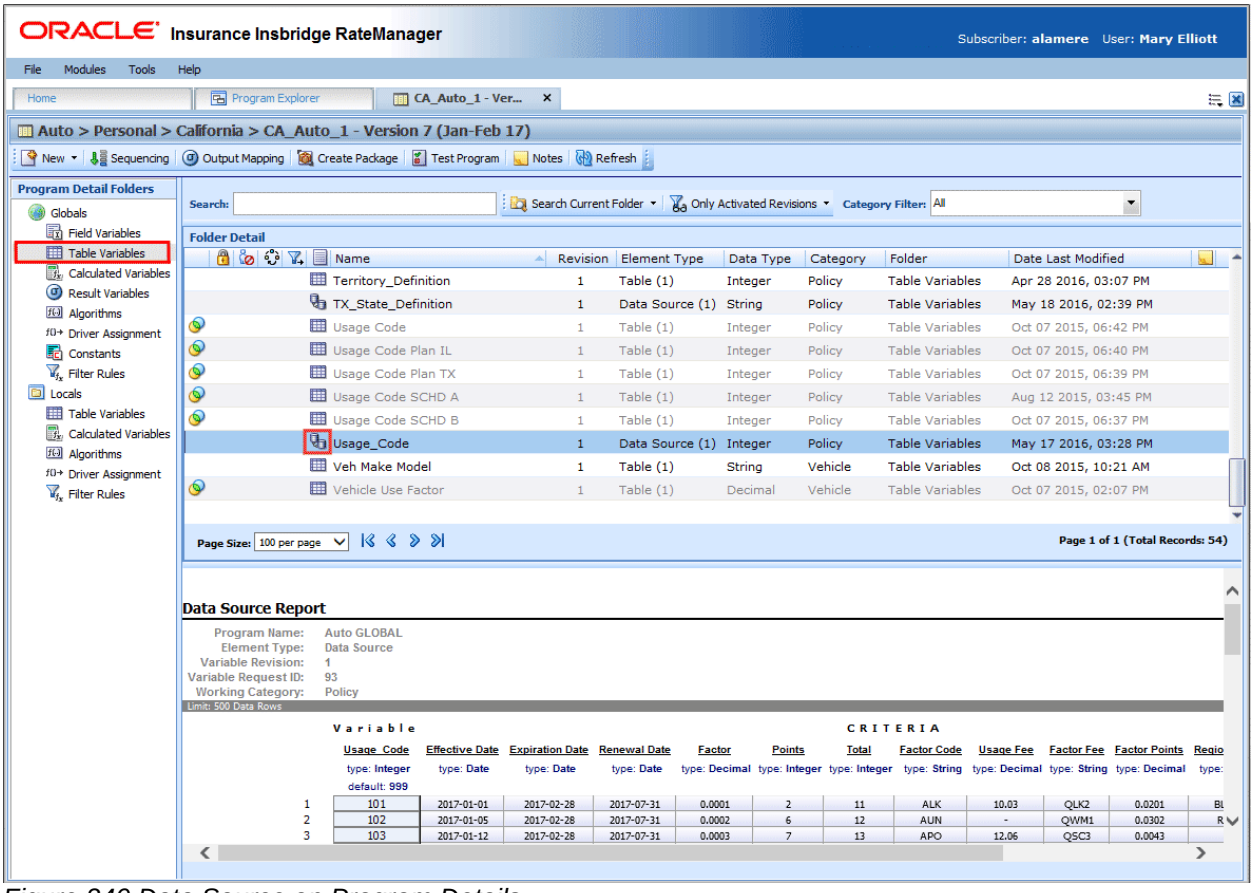


Figure 240 Data Source on Program Details

Data Sources Grouping

Grouping indicates that the qualifier fields are related and may be entered in any order. Grouping the qualifiers together reduces redundant table rows and makes tables easier to manage. Grouping allows users to perform SoftData table lookups similar to a SQL "AND" command.

For example, in a table with 5 criteria columns, the first three columns may be from the same value type/source, but not in a specific order. That is - the range values may be the same -- A and B and C -- but not in a specific order -- B and A and C, C and A and B, etc. … but the result is always the same, regardless of the order of the first three criteria values.

To access the specified columns as a group of data, reduce redundancy and more easily manage table data, grouping can be added. Grouping indicates that the fields can be entered in any order. In the lower example, the table data would effectively be reduced from 12 rows to 2 rows (group "A and B and C" and group "X and Y and Z").

Criterion1	Criterion2	Criterion3	Criterion4	Criterion5	Result1	Result2
String	String	String	Integer	Integer	String	String
A	B	C	1	2	No	Yes
A	C	B	1	2	No	Yes
B	A	C	1	2	No	Yes
B	C	A	1	2	No	Yes
C	A	B	1	2	No	Yes
C	B	A	1	2	No	Yes
X	Y	Z	8	9	Yes	Yes
X	Z	Y	8	9	Yes	Yes
Y	X	Z	8	9	Yes	Yes
Y	Z	X	8	9	Yes	Yes
Z	X	Y	8	9	Yes	Yes
Z	Y	X	8	9	Yes	Yes

Figure 241 Standard Criteria

When grouping is applied, the table can be reduced. At runtime, the engine sorts the criteria and then performs the table lookup.

Criterion1	Criterion2	Criterion3	Criterion4	Criterion5	Result1	Result2
String	String	String	Integer	Integer	String	String
GROUP						
A	B	C	1	2	No	Yes
X	Y	Z	8	9	Yes	Yes

Figure 242 Grouped Criteria

Grouped columns must be of the same data type. Grouping must be exclusive of criteria which are defined as "Interpolate" and/or "Wildcard".

Adding Grouping

Grouping is done on the criteria columns. Grouping indicates that the qualifier columns are related and may be entered in any order on the table.

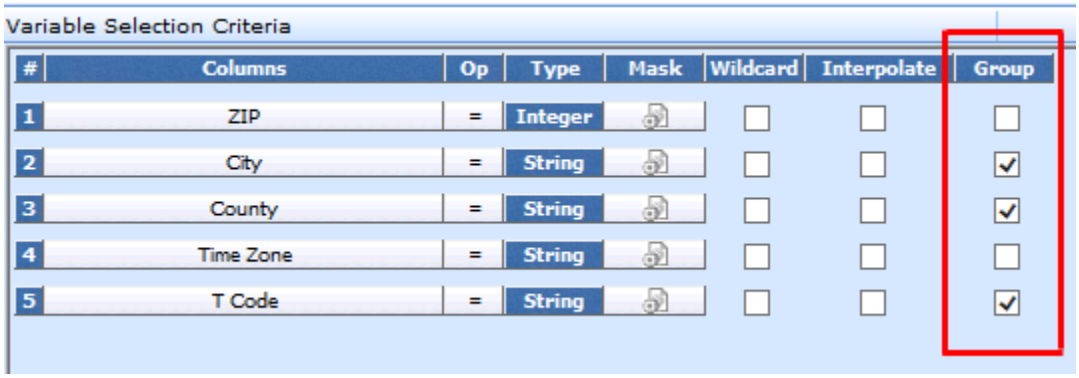


Figure 243 Data Sources Grouping

When adding grouping to a data source:

- Group members must be the same data type.
- Group members cannot have wildcard, or interpolation added.
- The group must have more than one member.
- Only one group is created per data source. For example, you cannot create a group of criteria for data type integer and then a second group for data type string.

Grouping can be selected after the criteria columns have been added. Check the group checkbox to add members to the group.

If the data type of a group member is changed, that member is removed from the group. If grouping is not needed, the grouping can be overridden on the SoftData call by setting the grouping attribute to zero.

Creating

When you create a new data source at the local level, it is available to the program only. If you create a new data source at the global level, it is available to any program in the product. If you create a new data source at the universal level, it is available to any program in the project.

Data sources are not used in any variables. Data sources can only be added to the output mapping of a program. Since data sources can only be used in SoftData queries, category does not matter. It is recommended that you leave the category at the policy level. Another category can be selected if desired.

Criteria column names must be unique per data source. Up to 50 criteria columns can be used.

1. To create a universal data source, select the universal icon under the project where you want to create a data source on Program Explorer. Click New, select Table and then select Data Source. To create a global or local data source, enter the program details tab from the program where you want to create a data source. In the top bar menu, select New, select either Globals or Locals, select Table and then select Data Source.

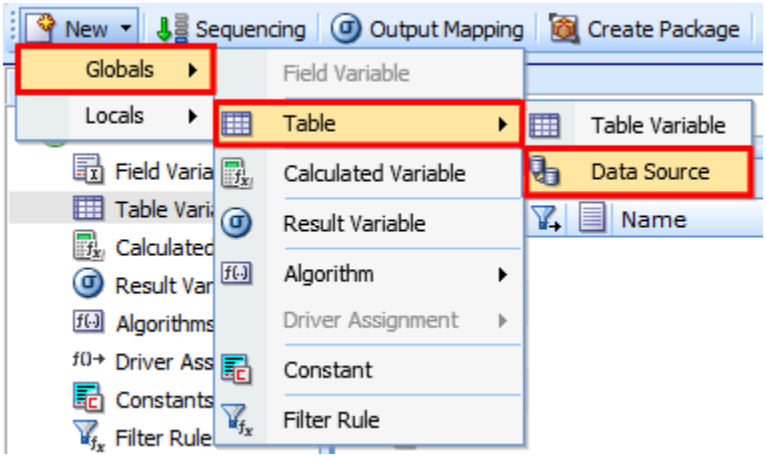


Figure 244 Navigation to New Data Sources

2. Enter in a **name**. Alphanumeric characters are permitted, but no special characters (ex: &, *, ", +, @, etc.) are allowed. In the case of linked data sources, each data source must have a unique name. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

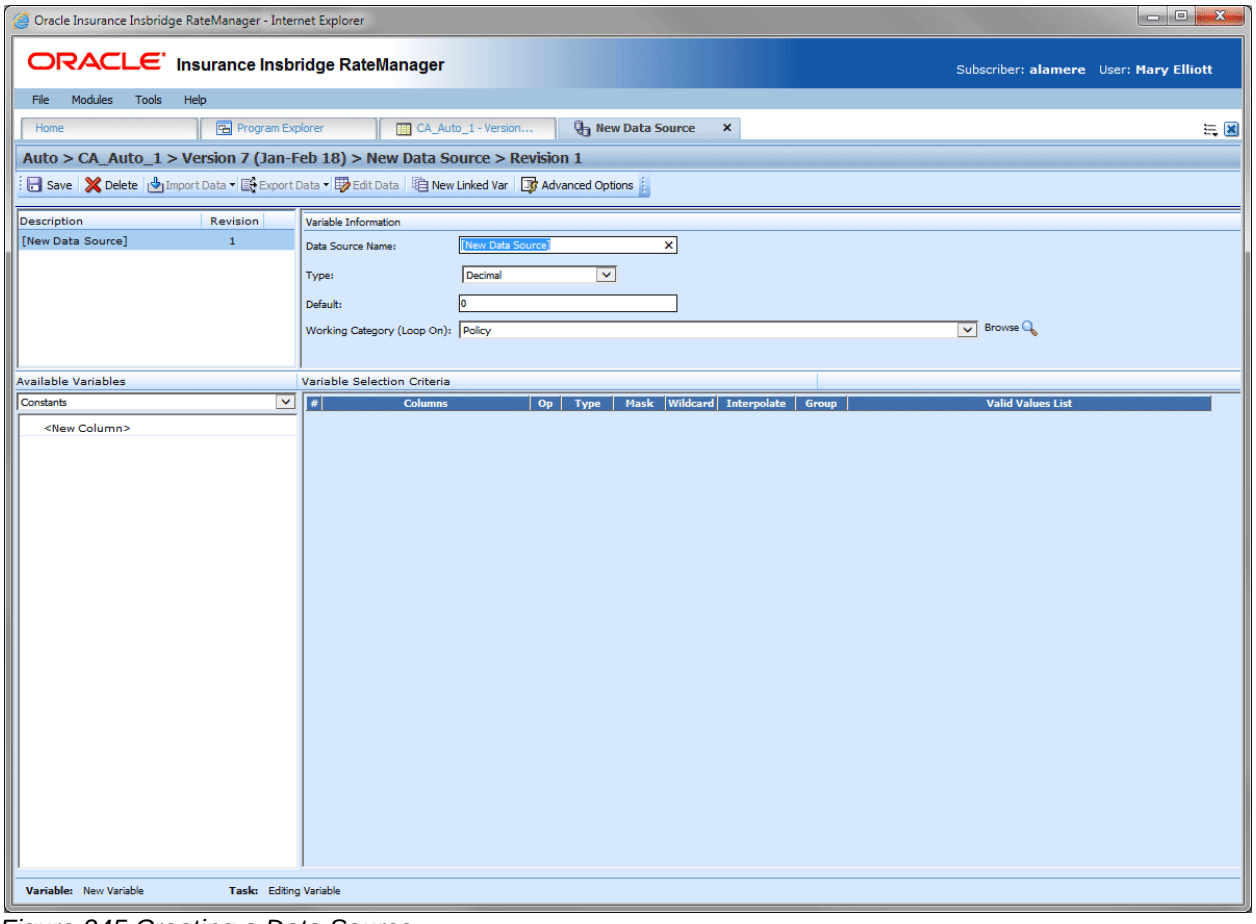


Figure 245 Creating a Data Source

3. Select the **type** of data associated with the variable. Options are:

- **Integer**
- **String**
- **Decimal**
- **Date**

In the case of linked variables, a different data type may be set for each. If disabled by the system administrator in the configuration file, the data type cannot be changed for any existing variables.

4. Set the default value of the variable. The Default Value must match the data type selected and the field is limited to 50 characters.

5. If desired, change the working category from Policy.

6. In the Available Variables area, click <New Column>. Enter in a Column Name and select a data type. Click Save to save your entry and close the screen. You can enter up to 50 criteria columns.

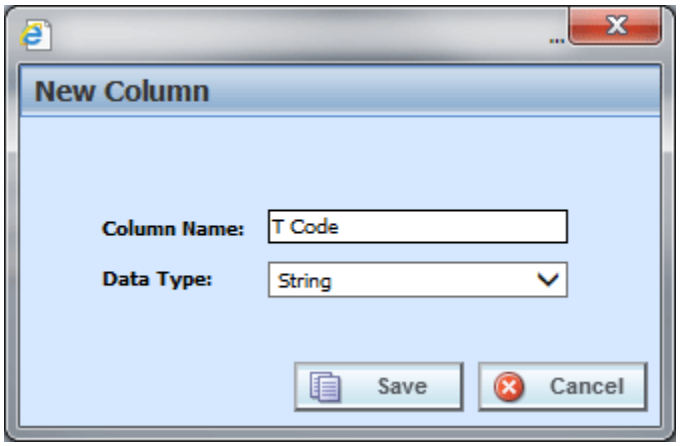


Figure 246 New Column for Data Source

7. If you want to apply a mask, click the mask icon. See Masking for more information.

8. Check the wildcard box for any variable where you want the system to accept any passed criteria. See Wildcard for more information.

9. Check the interpolate box if you want to use interpolation to determine the correct value for the table variable. See Interpolation for more information.

10. Check the criteria columns to be grouped. See Data Sources Grouping for more information.

11. Click Save to save your work.

Importing Data Sources Data

Data sources import and export functionality follows the same rules and processes as table variables. Please see Working with Table Data on page 287.

EDITING DATA SOURCES DATA

Data sources data can be edited directly in the table. You select a row where you can edit the data in an individual cell. Data within an individual cell can be added, deleted or updated. Single or multiple rows can be added or deleted. For data sources that are locked, data can be viewed in the data preview area of the Program Details tab but changes are not allowed. A new revision can be created that will be open for editing.

If a lot of changes are needed on a data source, you may want to export the data source, edit it, and then re-import the table. Be aware that database row numbers may be updated.

NOTE: *If you are on the **Edit** tab and you have changed the criteria information or added/deleted linked variables, remember to save your data source before editing the data.*

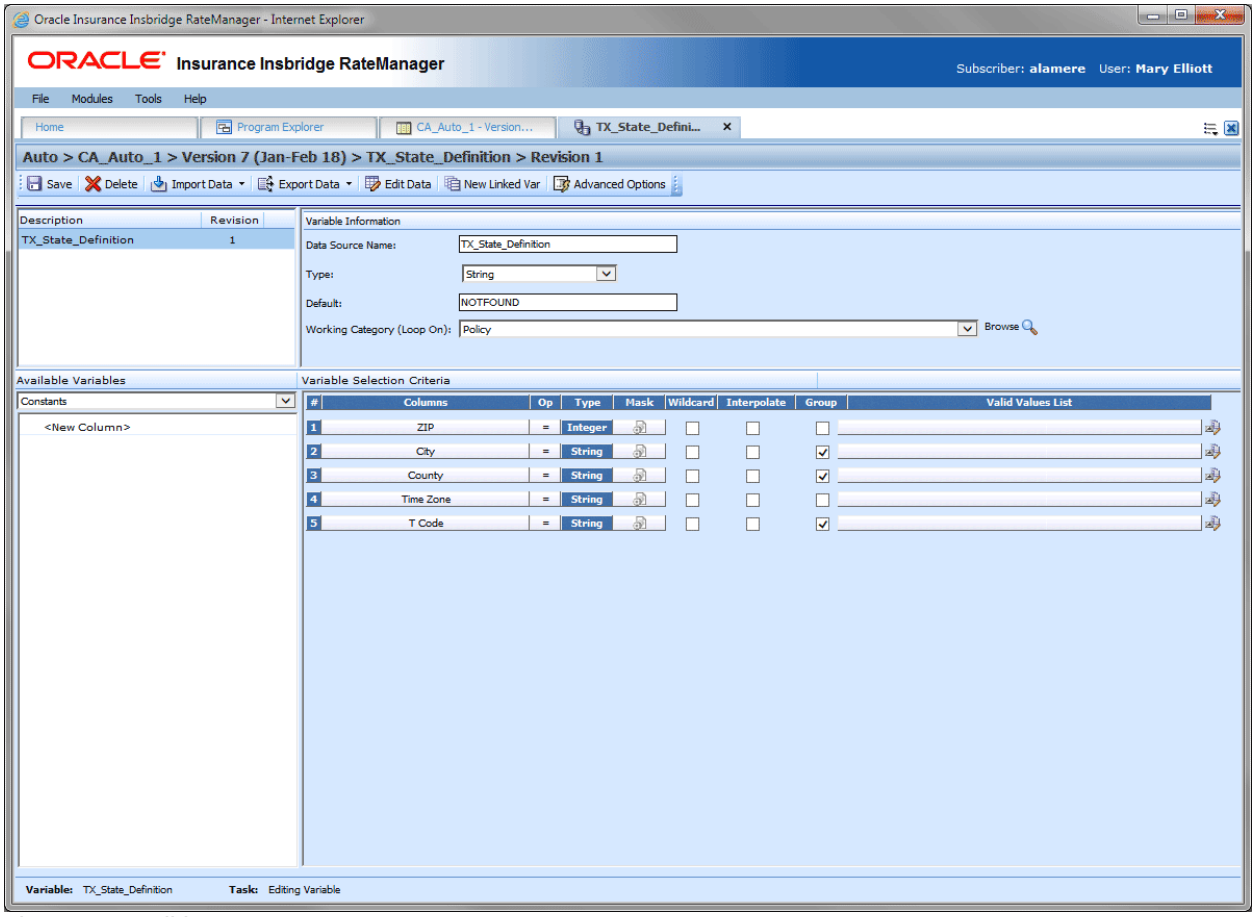


Figure 247 Editing Data Source Data

The Edit Data option on the Edit Variable tab opens the Editing Data screen. Editing rows or cells begins on the Editing Data screen.

Navigating Edit Data Screen

The Editing Data screen displays the current data source data. Editing and navigation options are at the bottom of the screen. This screen must be closed before viewing or working on any other tab in RateManager.

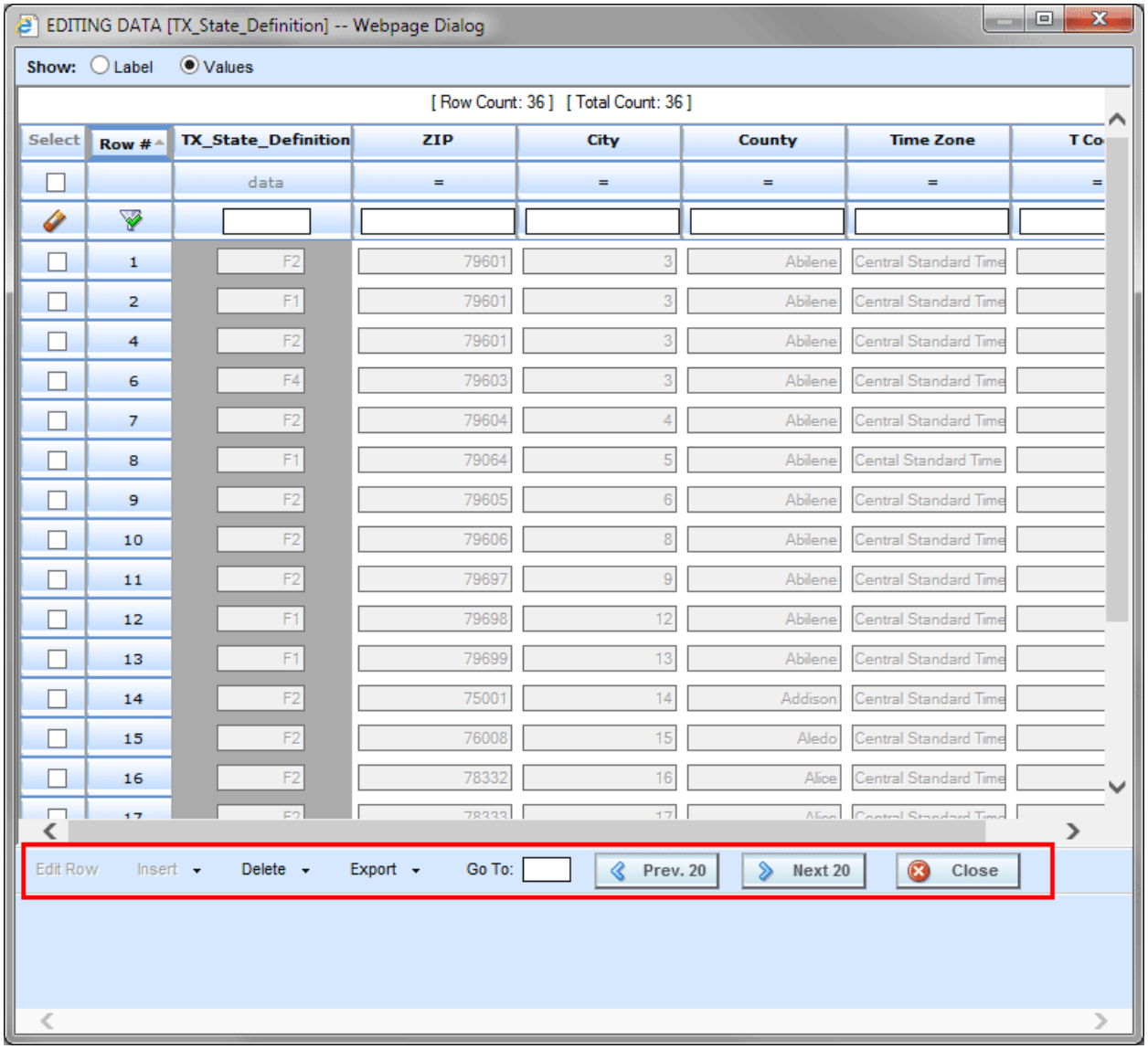


Figure 248 Navigating Editing Data Screen

Row Options

A row count for the entire table is listed at the top of the screen. Row numbers at the front of the row reflect the database row. The row count is updated as rows are added or removed. The database row number does not change when rows are removed. When rows are added, a database row number is assigned to the new row. Rows following the new row may have the database row number updated also. All database row numbers are updated when the entire table is re-imported. Database row numbers may be used in SoftData calls.

The button options at the bottom of the screen are for rows only.

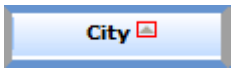
- Edit Row – Opens the Edit Row screen where edits can be made to the cells.
- Insert – Inserts a row either above or below the selected row.
- Delete – Removes the select row, or selected rows, or all rows on the page.
- Export – Exports the filtered data in either tab delimited or MS Excel format.

Paging

If the table contains more than twenty rows, use the **Prev.20** and the **Next 20** buttons to navigate through the table. You also can go directly to a row by entering the table row number in the **GO TO box** and clicking anywhere outside the box.

Icons and Options

The Edit Data screen contains icons and options to assist with editing table data.

Icon	What it Indicates
	Clear – Clears the filter criteria.
	Filter – Filters the entire table. The rows returned are only the rows that meet the filter criteria.
	Filter Removed – Removes the filter criteria.
	**Sort** – Sorts table data by the selected column in ascending or descending order. The column name is highlighted and the ascending or descending arrow is displayed.
	Show – When working with field variables that use a list of values, users can select the view for the valid values column. If there are no valid values, the view does not change.

Editing Data in a Data Source

Rows can be inserted or deleted on the main Editing Data screen, but data is edited on a separate screen Editing Data screen.

1. Navigate to the **Edit Variable** tab for the data source where you want to edit data.
2. Click **Edit Data** to open the editing screen.

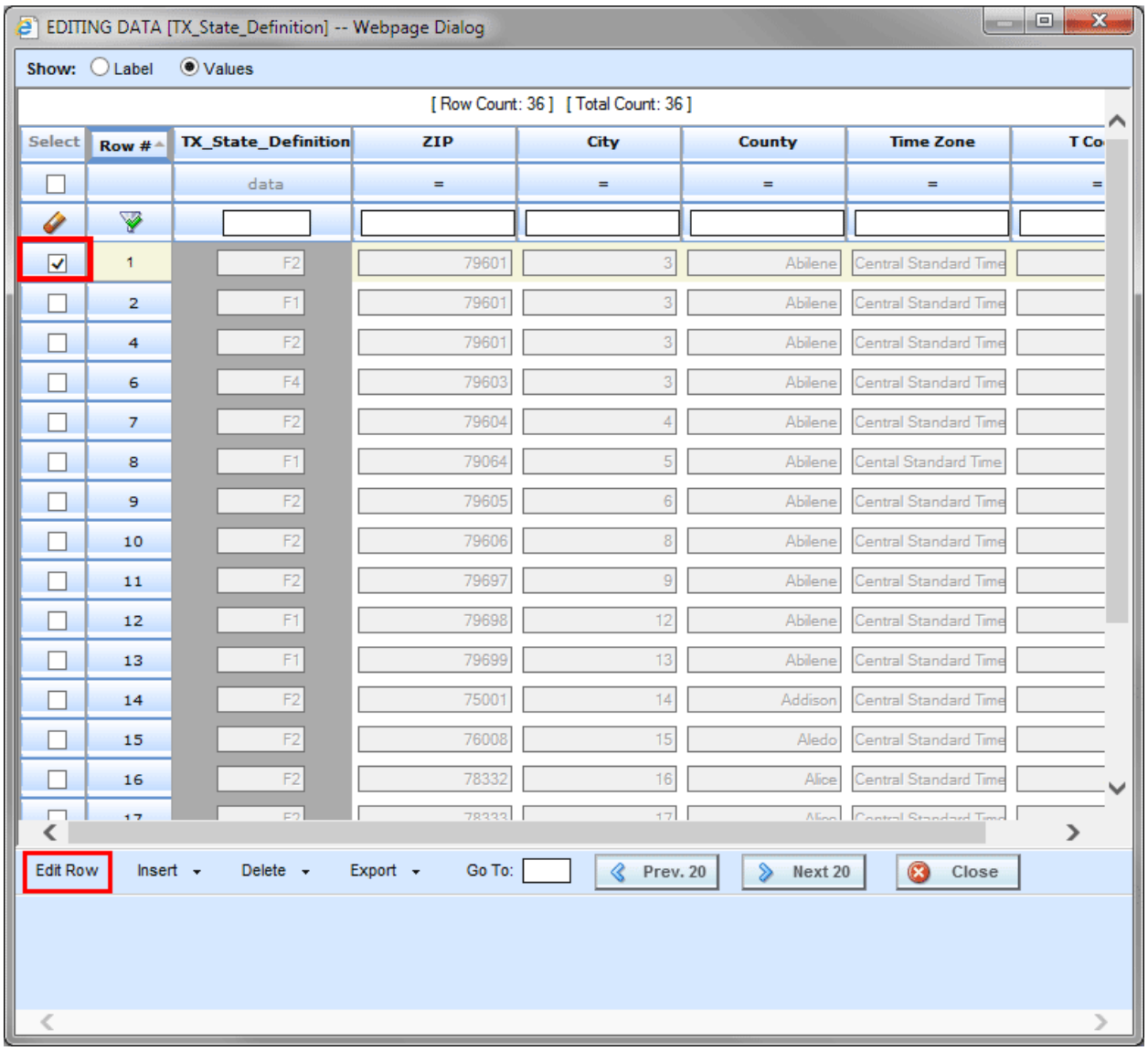


Figure 249 Editing Data Sources Data

3. Locate the row where you want to make edits. Check the Select option at the front of the row.
4. Click Edit Row. A separate editing screen opens.

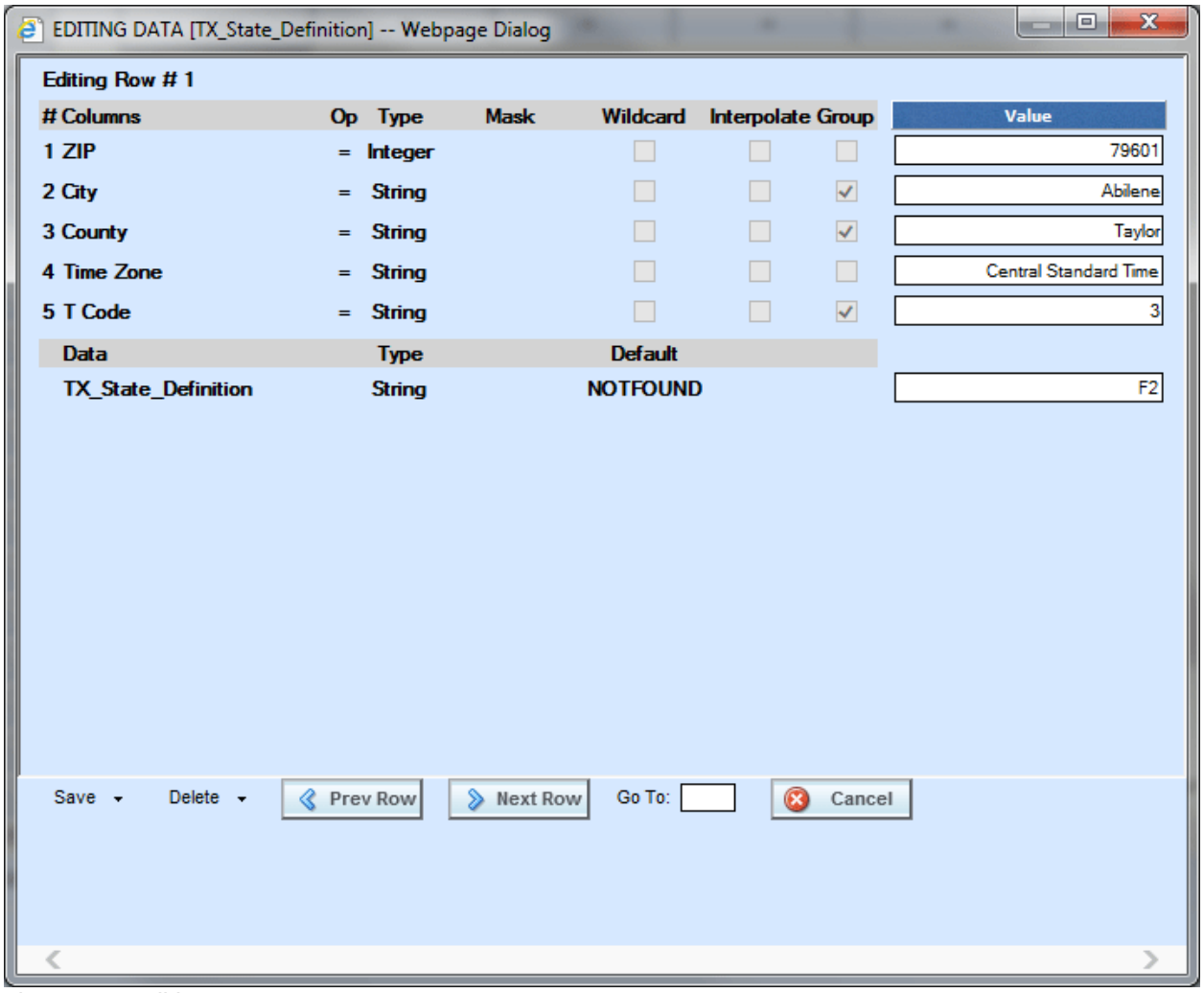


Figure 250 Editing Data Sources

5. Make any changes to the cell values. Cells can be tabbed through or the Enter key will move the cursor to the next cell. Mask, wildcard and interpolation along with criteria column information cannot be edited here. If you need to make changes to the criteria, return to the Edit Variable tab.

6. When you are finished editing the data, **Save and Close** to return the main **Editing Data** screen.

NOTE: *When you make a change to a cell, the data is **not** saved immediately. Changes to the data table must be saved before you leave the screen. No warning message is given if you have not saved the data. Clicking **Cancel** will close the Editing Data screen without saving data.*

Right Click Menu

Cells in the data table have a right click menu where you can select all contents in the cell, copy, cut, paste, or delete values. To use a right click menu option, select the cell where you want to edit data and right click.

Saving Data

There are three save options when editing data source data. These options allow you to continue working in the table:

- **Save and Next** – Saves the entries and opens the next row on the table for editing.
- **Save and New** – Saves the entries and creates a new row on the table directly under the row being edited.
- **Save and Close** – Saves the entries and closes the Editing Data screen.

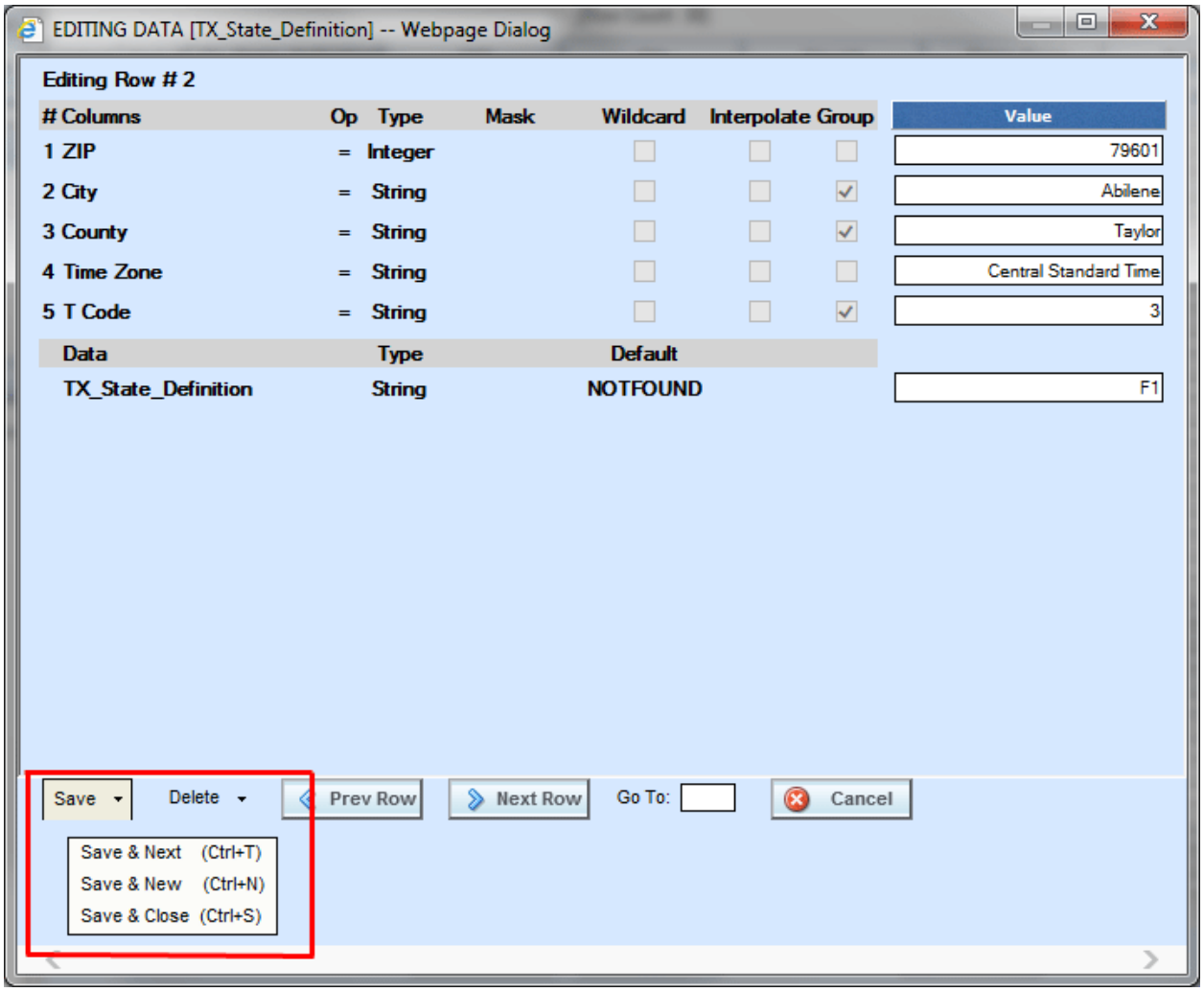


Figure 251 Saving Data on Editing a Data Source

You can click the option you want or use keyboard shortcut keys.

- **Save and Next** – Control Key and T. (Ctrl + T)
- **Save and New** – Control Key and N. (Ctrl + N)
- **Save and Close** – Control Key and S. (Ctrl + S)

Deleting Rows

There are two delete options:

- Delete and Next – Deletes the row and opens the next row on the table for editing. The deletion is saved.
- Delete and Close – Deletes the row and close the Editing data screen. The deletion is saved.

Make sure this is the action you want to take. Deletions cannot be undone.

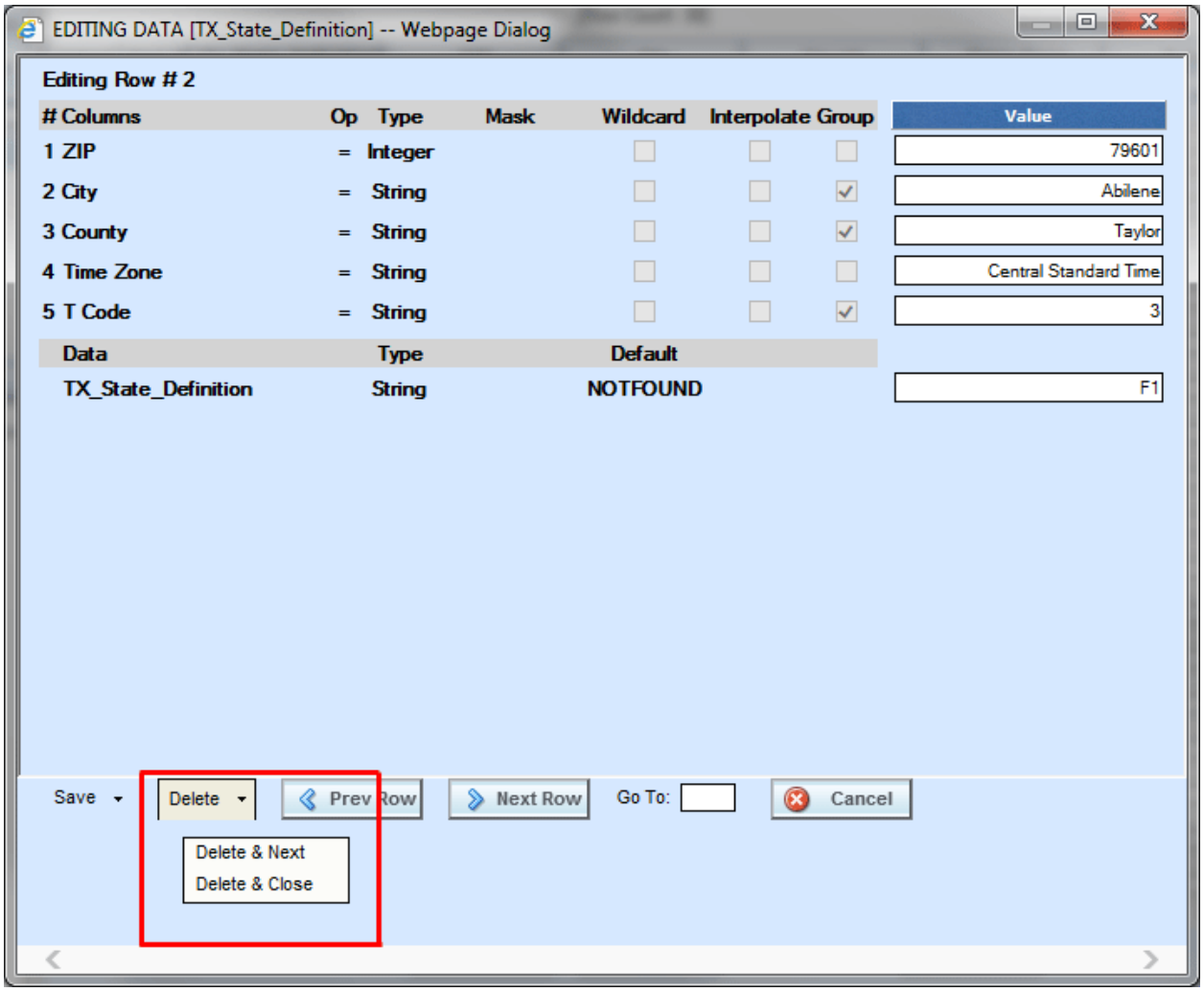


Figure 252 Deleting Rows from Data Source

NOTE: *Navigating through the table can be done with the Prev Row and Next Row option. Cell data changes are NOT saved when using the Previous or Next option. Make sure you do not have any changes to save before using Next or Previous.*

Editing Rows of a Data Source

Rows can be inserted, or deleted. The Insert and Delete options are on the lower left portion of the screen, next to the Previous 20 button.

1. Navigate to the **Edit Variable** tab for the data source where you want to edit data.

2. Click **Edit Data** to open the editing screen.

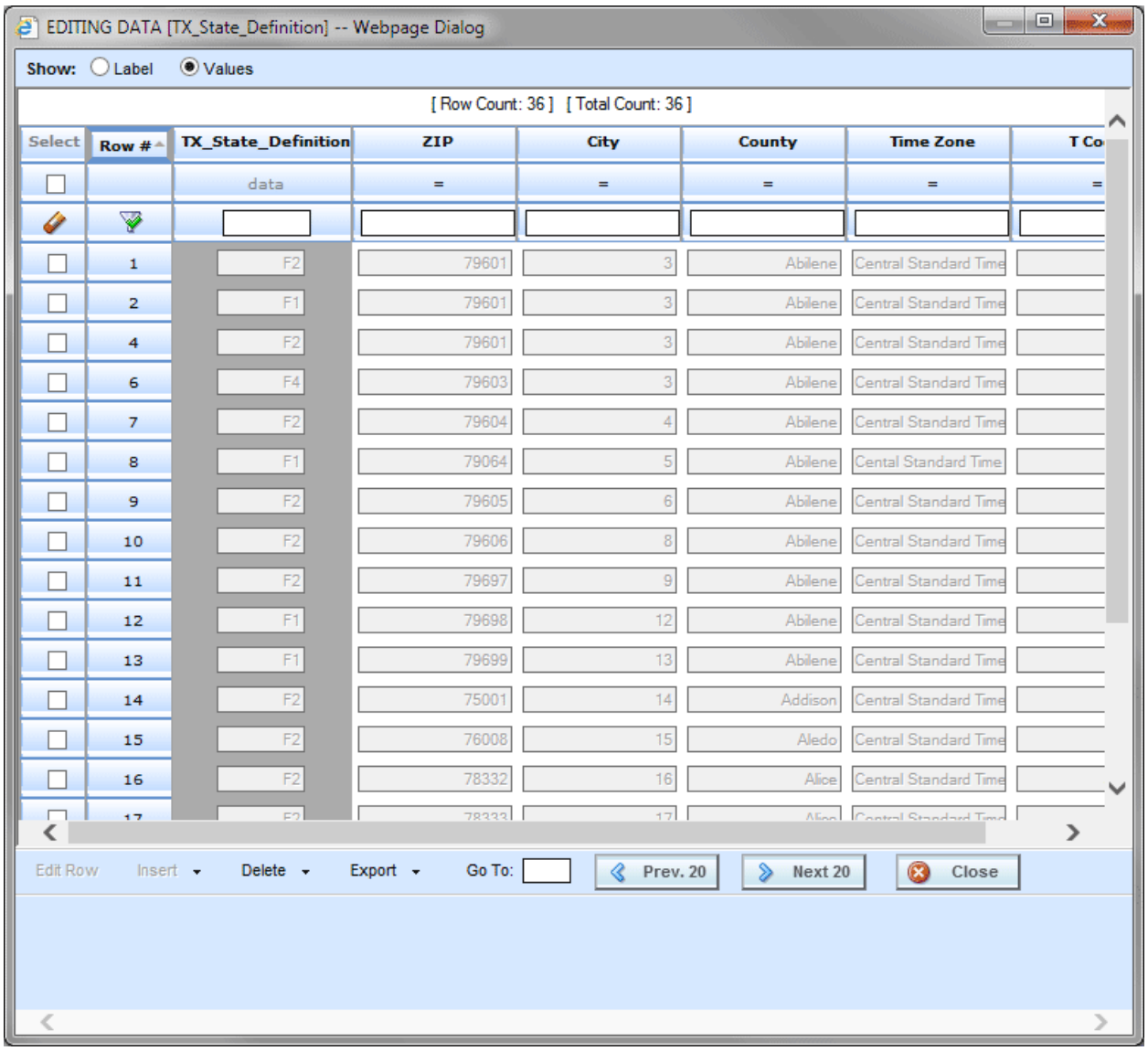


Figure 253 Editing Data Sources Data

3. Locate the row to be edited. If the table contains more than twenty rows, use the **Prev.20** and the **Next 20** buttons to navigate through the table. You also can go directly to a row by entering the row number in the **GO TO box** and clicking anywhere outside the box.

4. When you are on the row you want, you can insert, or delete.

Inserting Rows

Rows can be inserted any place in the table. A specific row must be selected where the new row is to be inserted. You can only insert one row at a time.

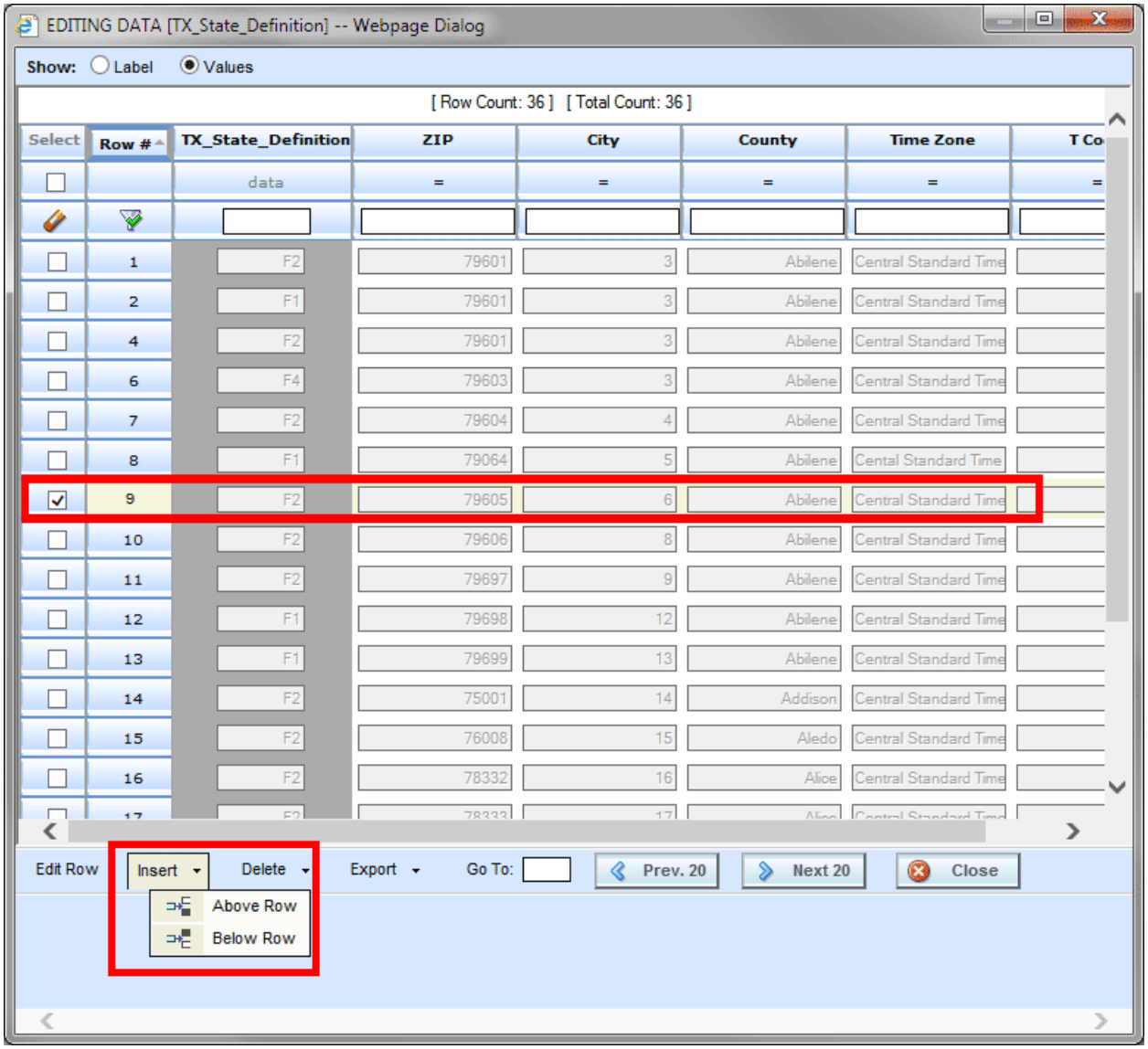


Figure 254 Inserting Rows on Data Source

1. To insert a new row, check the Select option at the front of the row where you want to insert a new row.

2. Click **Insert** and then select either **Above Row** or **Below Row** to insert the new row either above or below the selected row.

Deleting Rows

A single row or multiple rows can be deleted from the table. There is a warning message prior to the row or rows being deleted. Deletions are per page. You cannot select multiple rows across multiple pages and delete. Only the rows on the page you are currently on can be deleted. Make sure this is the action you want to take. Deletions cannot be undone.

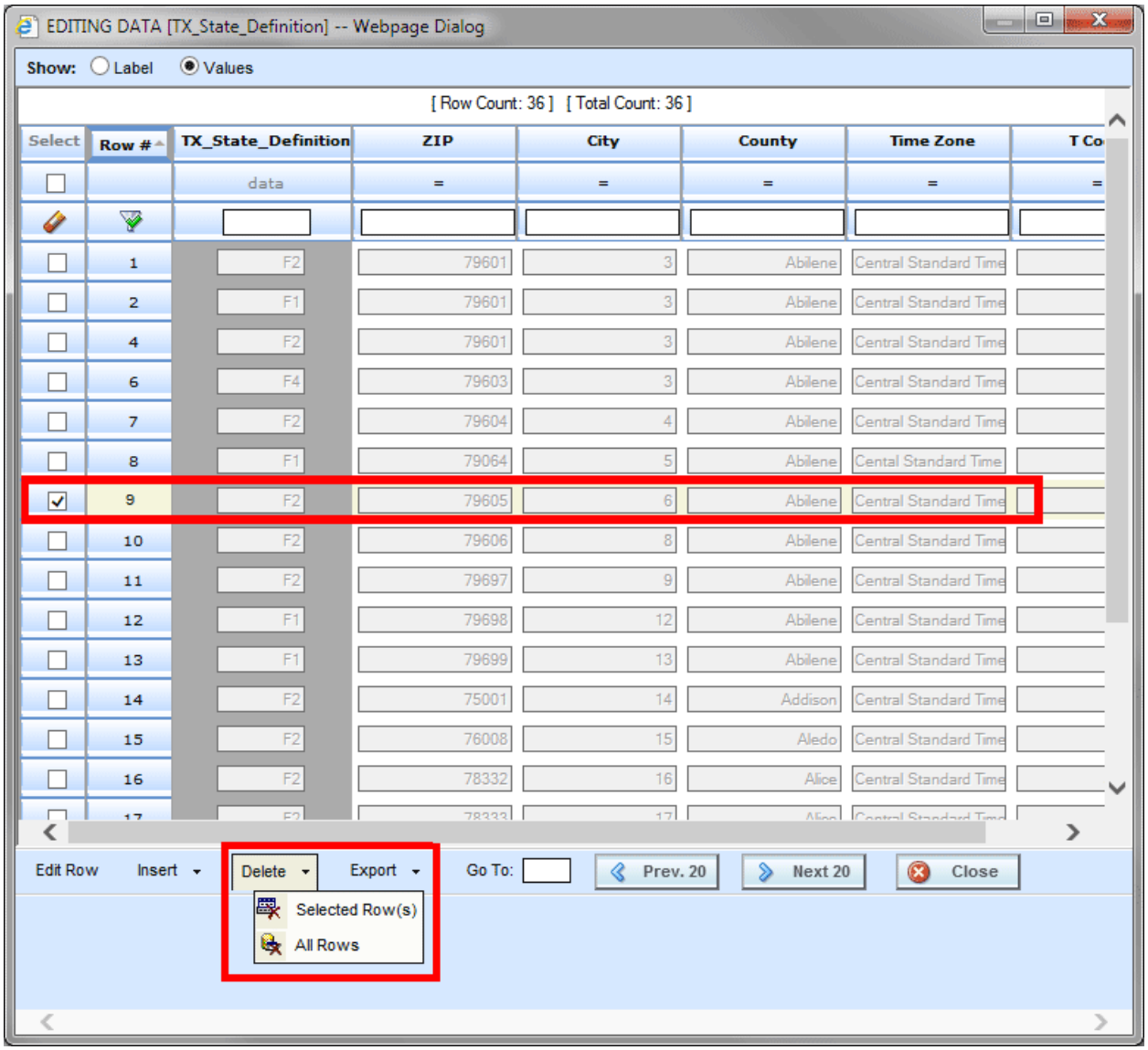


Figure 255 Deleting Rows from Data Source

1. To delete a single row from the table, check the Select option at the front of the row that you want to delete. Click **Delete** and **Selected Row(s)**. You will be asked to confirm deleting the row.

2. To delete more than one row, check the boxes in the Select column of the rows you want to delete. Verify your selections. Click **Delete** and **Selected Row(s)**. You will be asked to confirm deleting all rows.

3. To delete all rows from the table, click **Delete** and **All Rows**. You will be asked to confirm deleting all rows.

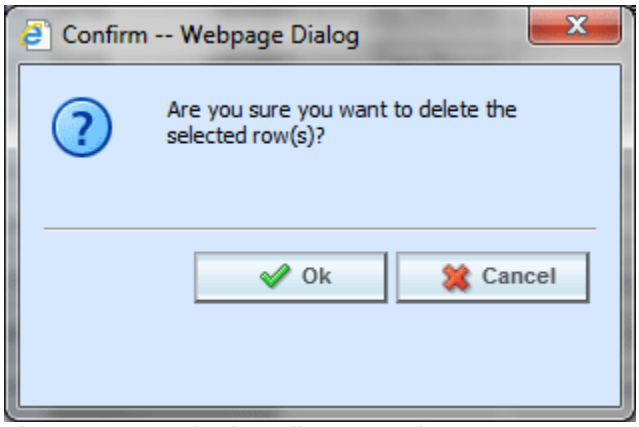


Figure 256 Confirming All Rows Delete

4. Select **Ok** to delete selected rows or all rows and return to the Edit Data screen or **Cancel** to return to the Edit Data screen without deleting any rows.

Filtered Rows

Data Source table data can be filtered to make it easier to locate or edit data within RateManager. Filters can be added to any row in the table and the data returned is only the data that matches the filter criteria. The match must be exact. The filter is done on the entire table.

Data can be filtered by one or more columns. Additional criteria can be entered to further define the data returned. Filters can be removed to expand the data returned. If a filter is in place, any data that does not match to filter is not presented.

Filtered data stays in place until the filter is removed or updated. Filtered data can be sorted or exported. The filter is for editing purposes only and does not remove data from the database or table.

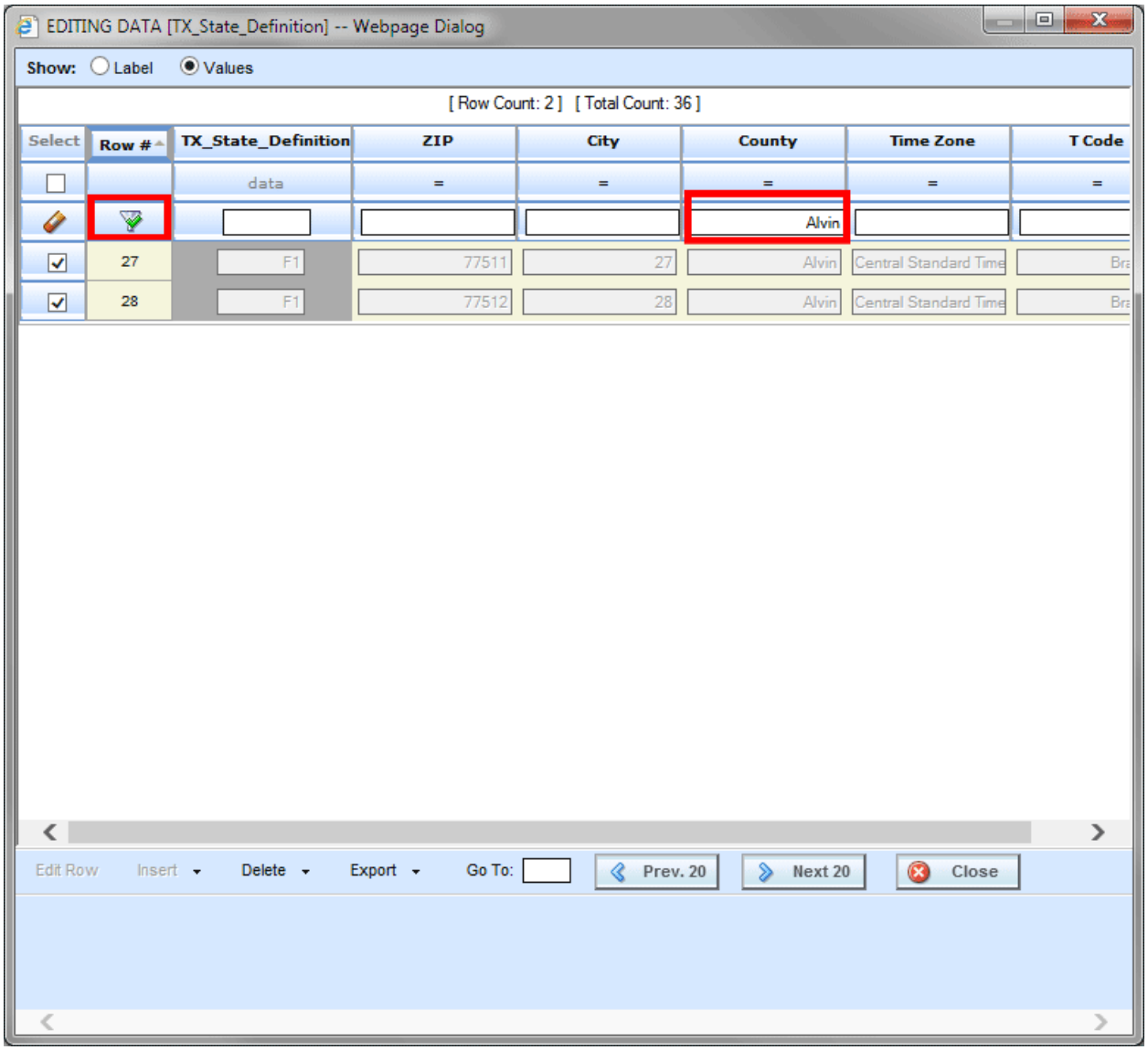


Figure 257 Filtered Data Source Data

1. To filter all the data in the table, enter filter criteria the field at the top of the column where filtering is to be done.

2. After entering a value, click the filter icon. The data returned is the data that meets the filter criteria. Additional criteria can be entered to further define the data returned.

Suspending a Filter

To suspend filtering, click the filter icon a second time. The icon is displayed with a red mark and the filter values are grayed out. This suspends all filtering. If you want to resume filtering or edit the filter criteria, click the filter icon again.

When filtering is suspended, the entire data table is returned and the row count is updated.

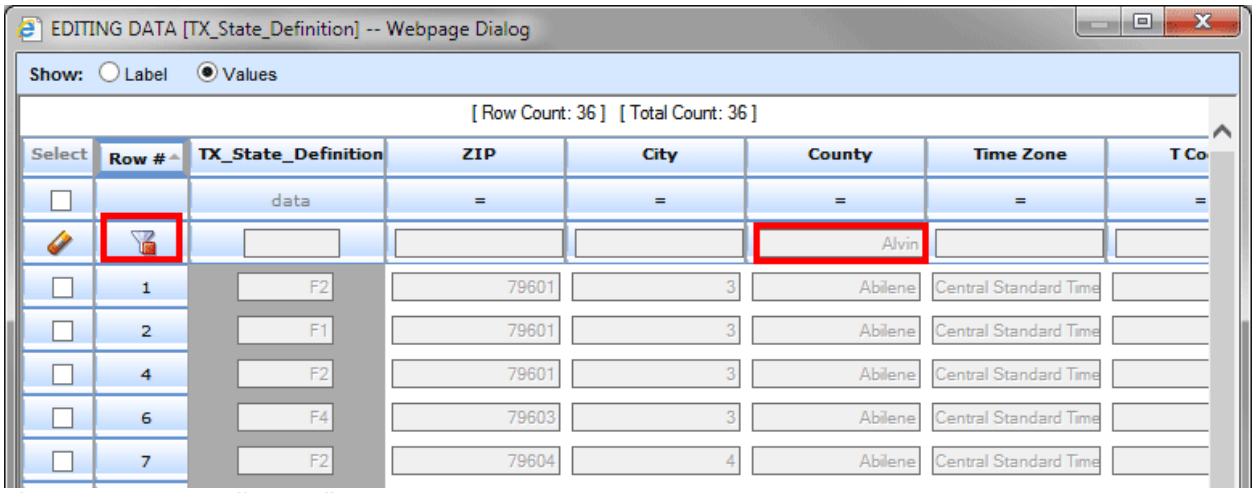


Figure 258 Suspending a Filter

Removing a Filter

To remove a filter, click the eraser icon . All filter criteria are removed from the top field and the entire data table is returned.

Filtering Data Source Data with a List of Values

When working with table data that contains a list of values, filtering is done on the value, not on the label. To make sure the desired results are displayed, select the values view in the Show option. A column with a list of values has a green checkmark and a list of values icon next to the data field.

Sorting Rows

Data source data can be sorted to make it easier to locate or edit data within RateManager. Data can be sorted by any column selected. The sort is done on the entire table.

Any single column can be used for sorting. The sort is applied to the entire table and remains in place when paging through the data or applying a filter.

The default sorting row is the row number. If sorting by another column is no longer needed, click the row number column to return to the default.

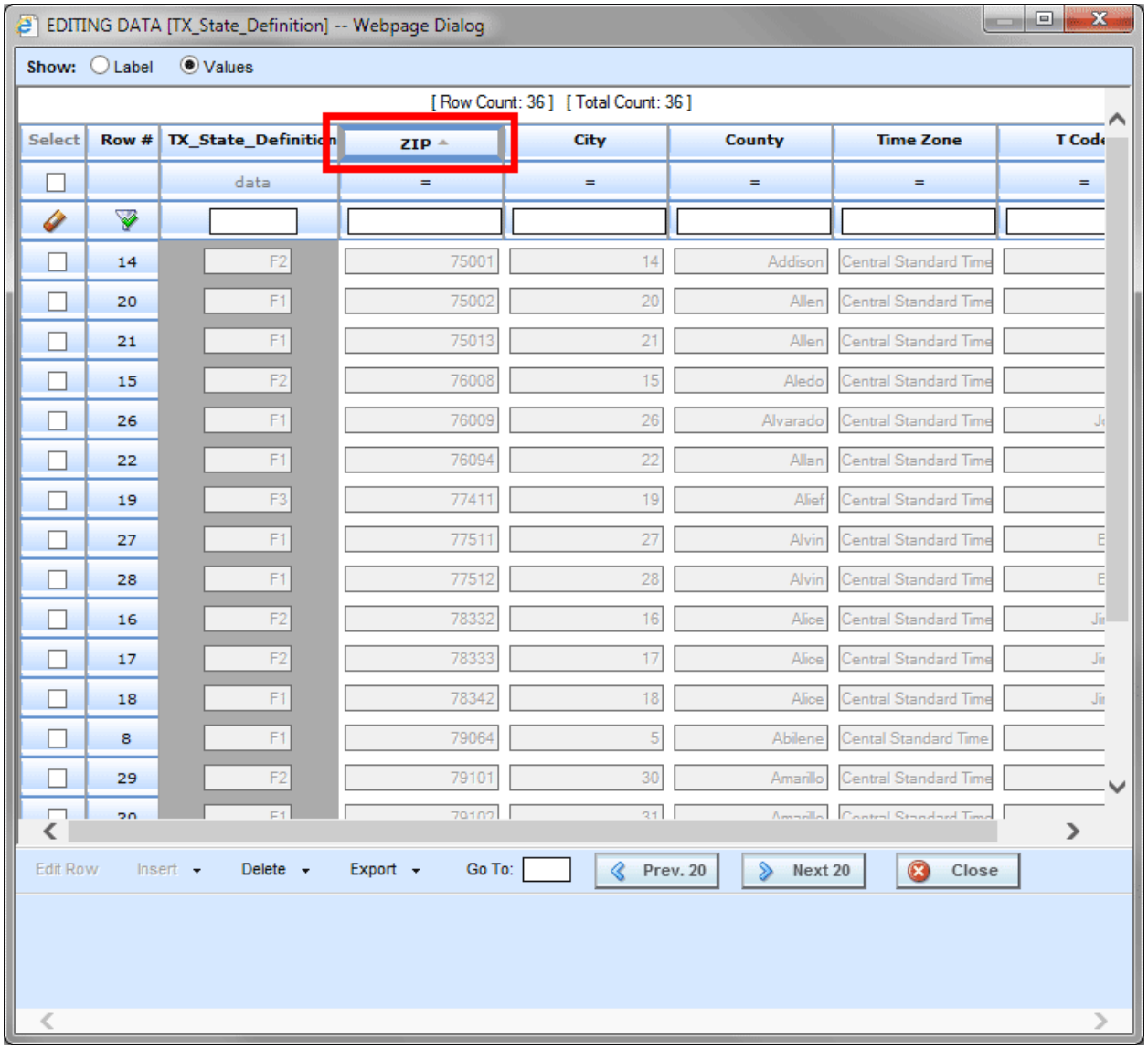


Figure 259 Sorted Data Source Data

- To sort by a specific column, click the column header. The header is highlighted and the sort arrow is displayed. Only one column determines the sorting.
- Sorted data can be in ascending or descending order.

- A downwards arrow indicated descending order, and up arrow is ascending order.
- Sorting can be done on data that is filtered or uses a list of values.

Exporting Rows

The Export option at the bottom of the edit data screen allows for filtered rows of data to be exported in tab delimited or MS Excel format. Exporting smaller sections of data can save time and reduce the risk of an accidental edits to data that did not change. Data can be filtered by one or more columns. The data returned is the data that meets the filter criteria. When the required subset of data is complete, the data can be exported in tab delimited format or MS Excel format.

For example, if there is a table with 100,000 rows and changes need to be made to 50 rows with common data values, table data can be filtered down and the filtered rows can be exported in MS Excel format. After any edits, the data can be re-imported and appended to the table.

The number of rows to be exported from the edit table data screen is listed at the top of the screen. All data that meets the filters is exported. Exports are not done per page.

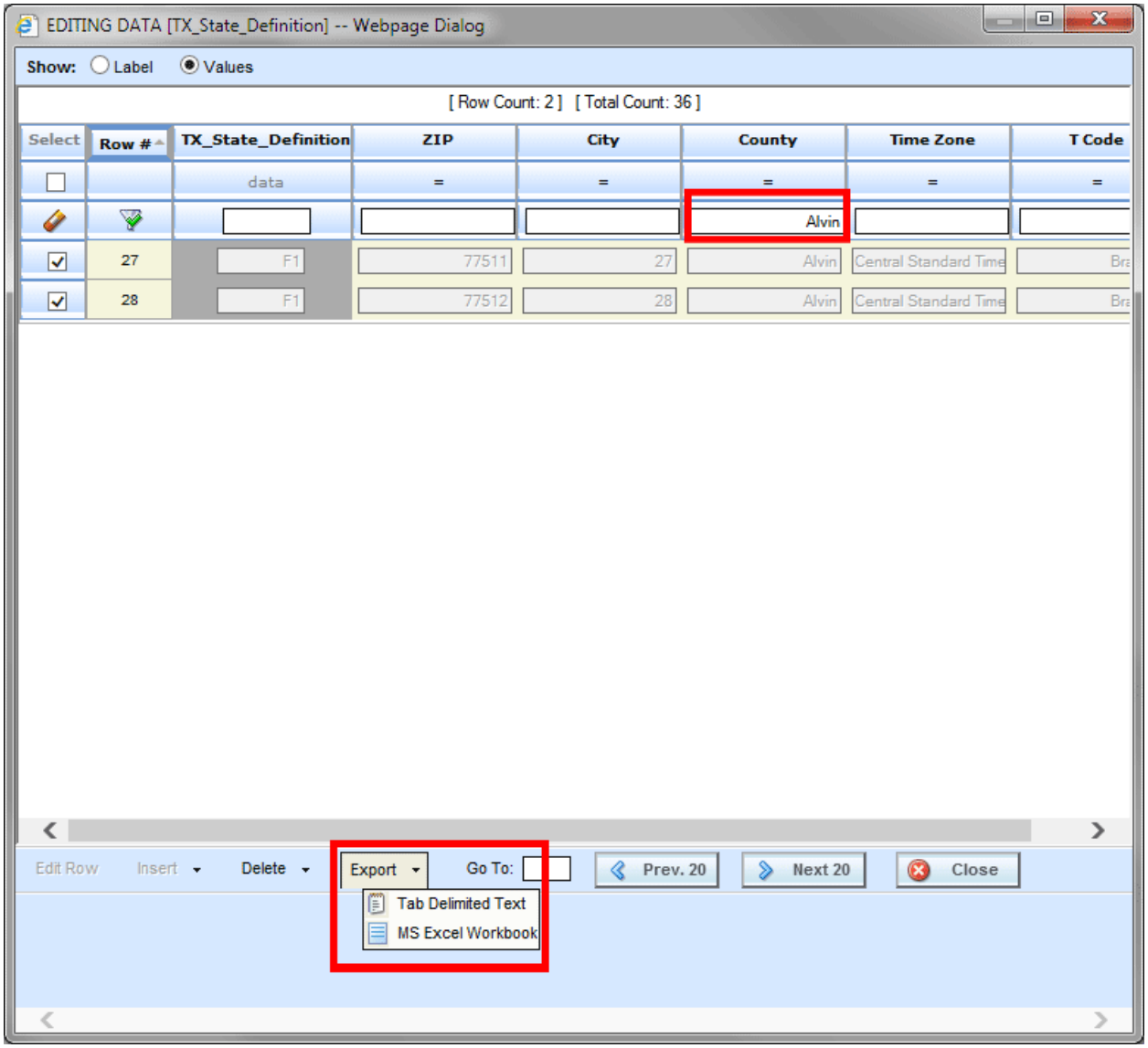


Figure 260 Filtered Data for Export

1. On the Editing Data screen, enter filter criteria for the rows or rows where you want to filter data.

2. Click the filter icon. The screen refreshes with the data that matches the filter. The number of rows returned is listed at the top of the screen.

3. When the subset of data is complete, select the type of export needed; tab delimited or MS Excel.

Tab delimited reports are presented immediately. MS Excel reports can be saved to a table job or run immediately. Please see Export Data to MS Excel Workbook for more information.

To allow for exports to MS Excel, the same setup procedures for table data exports must be in place.

Data exported in tab delimited format cannot be appended to the table.

DELETING A DATA SOURCE

When a data source is no longer needed, it can either be left in RateManager or deleted. When you delete a data source, it gets deleted from all versions of the program. Also, all revisions of that data source get deleted. If you are uncertain as to whether the data source will be used again, simply leave it in RateManager. This saves time if the data source is needed and does not slow down rating.

1. Select the data source you want to delete. Right click it and select **Delete**.

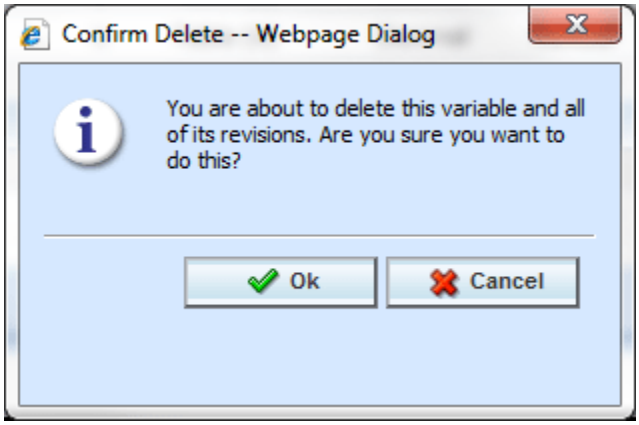


Figure 261 Deleting a Data Source

2. You will be asked to confirm the deletion of the data source. Click **Ok** to delete. Click Cancel to return to the tab without effecting a change.

Data sources that are in an output mapping can be deleted. Data sources that are locked cannot be deleted.

If you delete a linked data source, only the data source you have selected is deleted. The other data sources that are linked to it are not removed.

Working with Data Sources

Data Sources are used in SoftData Version 2 calls only. To use a data source in a SoftData call, you must add the data source to the output mapping for the program.

1. Select the program you want to use for the SoftData call.
2. Navigate to the Output Mapping screen.
3. On the output mapping screen, select Table Variables for the Variable Filter. Data sources are listed.

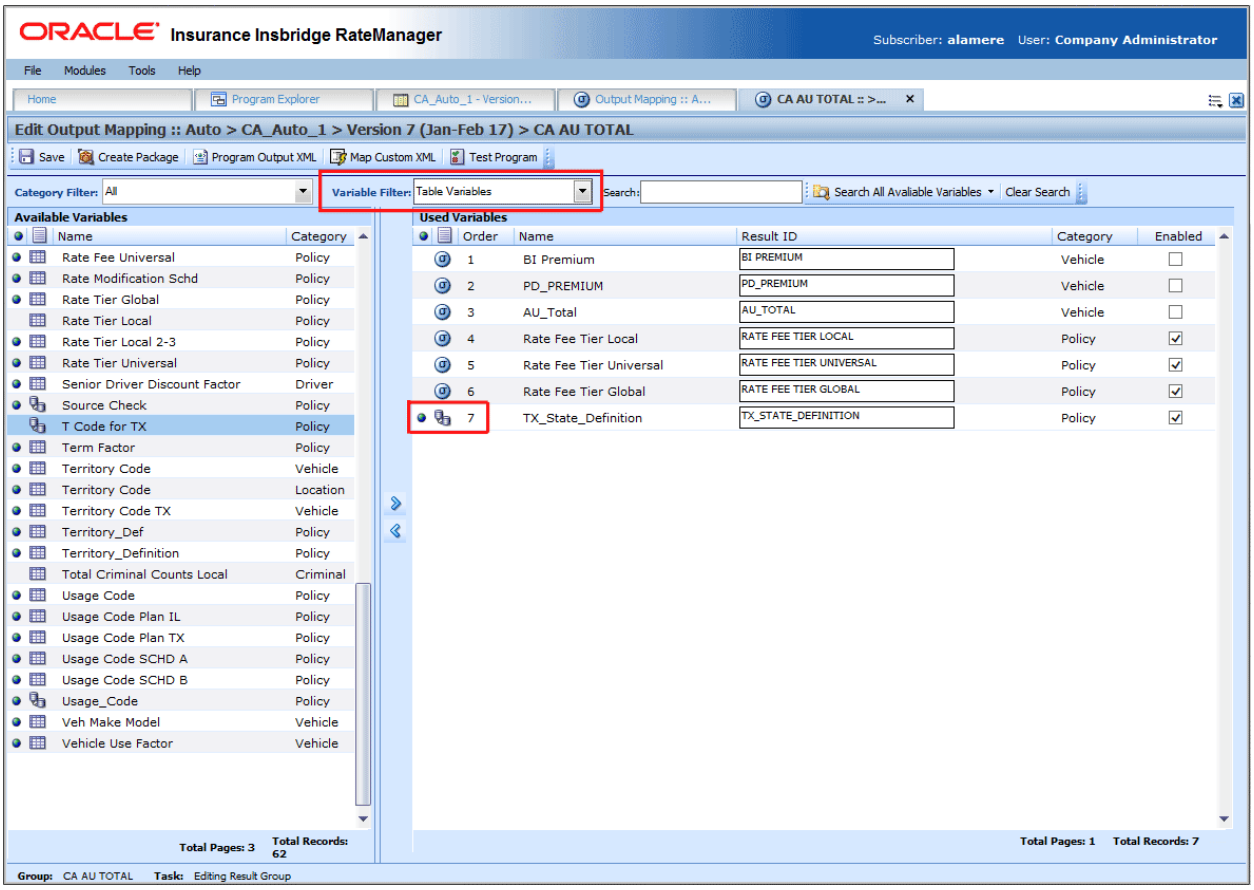


Figure 262 Data Source on Output Mapping

4. Select the data source you need and add it to the Used Variables.
5. Package and Test the Program.

Data sources are listed on Test Case Editor with the default value displayed.

Chapter 17

WORKING WITH STEPS

Steps are used use in algorithms, calculated variables, driver assignment algorithms, driver assignment scenarios, or filter rules. The following step types are available for every element:

- Category Items:
  - Get Category Item
  - Set Category Item
  - Get Ranked Category Item
  - Set Ranked Category Item
- Count Across
- DataType Functions:
  - IsNumeric
  - IsAlpha
  - IsDate
- Date Functions:
  - Date Addition
  - Date Difference
- If Statement
- Mask
- Math Functions:
  - Arithmetic
  - Absolute Value
  - Power
  - Log
  - Log10
  - Natural Log
  - Exp
  - Rand
  - Fact
  - Sqrt
  - Ceiling
  - Floor
  - Even
  - Odd
- Trig Functions:
  - Cos
  - Cosh
  - Acos
  - Acosh
  - Sin
  - Sinh
  - Asin
  - Asinh
  - Tan
  - Tanh
  - Atan
  - Atanh
  - Degrees
  - Radians
- Statistical Functions:
  - Gamma

  - Incomplete Gamma
  - Log Gamma
- Ranking:
  - Clear Ranking
  - Rank Across Category
- String Functions:
  - Get Length
  - Set Message
  - String Addition
- Sum Across
- Multiply Across
- Minimum/Maximum

The following step type is available only for use in underwriting algorithms:

- Set Underwriting to Fail

The following step types are available only for use in driver assignment scenarios:

- Set Principal Operator Variable
- Re-calculate Vehicle Usage: Using

The following step type is available only for use in local algorithms using a Create Category Instance function:

- Query Data Source

For all elements that use steps, you can set the next step to be executed, reorder steps, edit and delete steps.

NOTE: *In the information in this chapter the word algorithm is used as a generalization. The information applies to all algorithms, calculated variables, driver assignment algorithms, driver assignment scenarios, and filter rules.*

ENTERING A NEW STEP

New steps can be entered for any unlocked calculated variable, algorithm, driver assignment, or filter rule in a standard program. In a template generated program, any calculated variable, algorithm, driver assignment, or filter rule that is open for editing can have steps added. Whether you are editing an existing program or creating a new one, the process is the same.

1. Navigate to Program Details Tab for the program where you want to add a step.

2. Navigate to the element where you want to add a step. This can be a new element or an existing one.

3. Either double click the variable or select it and right click. Select **Edit** from the menu. This opens the **Edit Variable Tab**.

4. To add a new step, click **New Step** in the top bar menu and select the type of step you want to add from the menu.

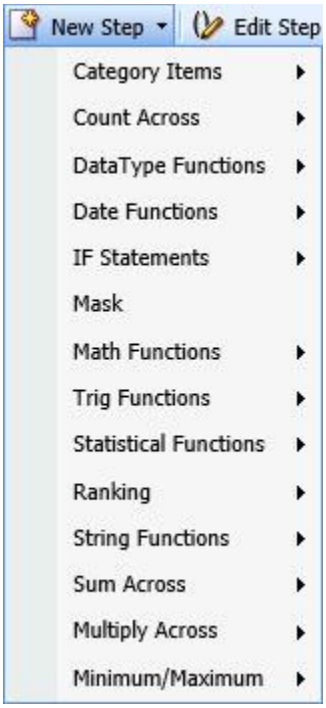


Figure 263 Entering a New Step

5. The new step type is added in the variable details area. Depending upon the step type, you can select the appropriate variables, constants, fields. The values needed may be listed on the step. For example, if the step has an Enter Variable box, you must choose the variable type from the Available Variable, Functions, and Constants area on the right side of the screen. The options are populated underneath. Click the Enter Variable box and then double click the corresponding variable you want to use. The box refreshes with your selected value.

6. Continue entering steps until complete. When you are finished making your entries, click **Save**.

7. After all steps have been entered, you can set the next step, update comparison operators, and add options to select steps.

8. When complete, click **Save** to save your work.

EDITING STEPS

Any step in the Steps list is open for editing. Steps can be:

- Copied and Pasted
- Assign the Next Steps
- Use Result of Steps
- Add Terms
- Add Rounding
- Define or Change Comparison Operators
- Add Where Clauses to select step types
- Add Conditions to select step types.

Prior to editing, you may want to move steps to the desired location and delete any step that you do not want. This can assist with setting the Next Step and Results of Step values.

COPY AND PASTE STEPS

Single or multiple steps can be copied and pasted in any unlocked calculated variable, algorithm, driver assignment, or filter rule in a standard program. In a template generated program, any calculated variable, algorithm, driver assignment, or filter rule that is open for editing can have steps copied and pasted.

Copy and pasting steps is done using the Edit Step button in the top menu of the element. The options are:

- **Copy Step** – Copies the selected step to the clipboard. This option is always available.

- **Paste Step** – Pastes the step(s) that are on the clipboard either above or below the selected step. This option is available when there are contents on the clipboard. If the clipboard is empty, this option is grayed out.

- **Clipboard** – Displays the currently copied steps in order. This option is available when there are contents on the clipboard. If the clipboard is empty, this option is grayed out.

Steps are copied one at a time. A single step is copied and placed on the clipboard. If more than one step is to be copied, they should be copied in the order in which they are to be pasted. Any step can be copied or copied multiple times if needed.

The entire clipboard is pasted either above or below a select step. If multiple steps were copied, the entire clipboard is pasted in the order in which the steps were selected. The pasted steps can be moved, edited or deleted.

Clipboard contents can be cleared after pasting. The contents of the clipboard can be viewed by selecting Clipboard->View. Clipboard contents can only be viewed; individual steps cannot be deleted or moved. The clipboard can be cleared without pasting by selecting Clipboard->Clear. Clipboard contents are held for as long as the element is open and are unique to the element. Clipboards are cleared when the element is closed.

Steps can be copied and pasted only within the current element.

1. Select the step that you want to copy.

2. Select **Edit Step**->**Copy Step** from the menu.

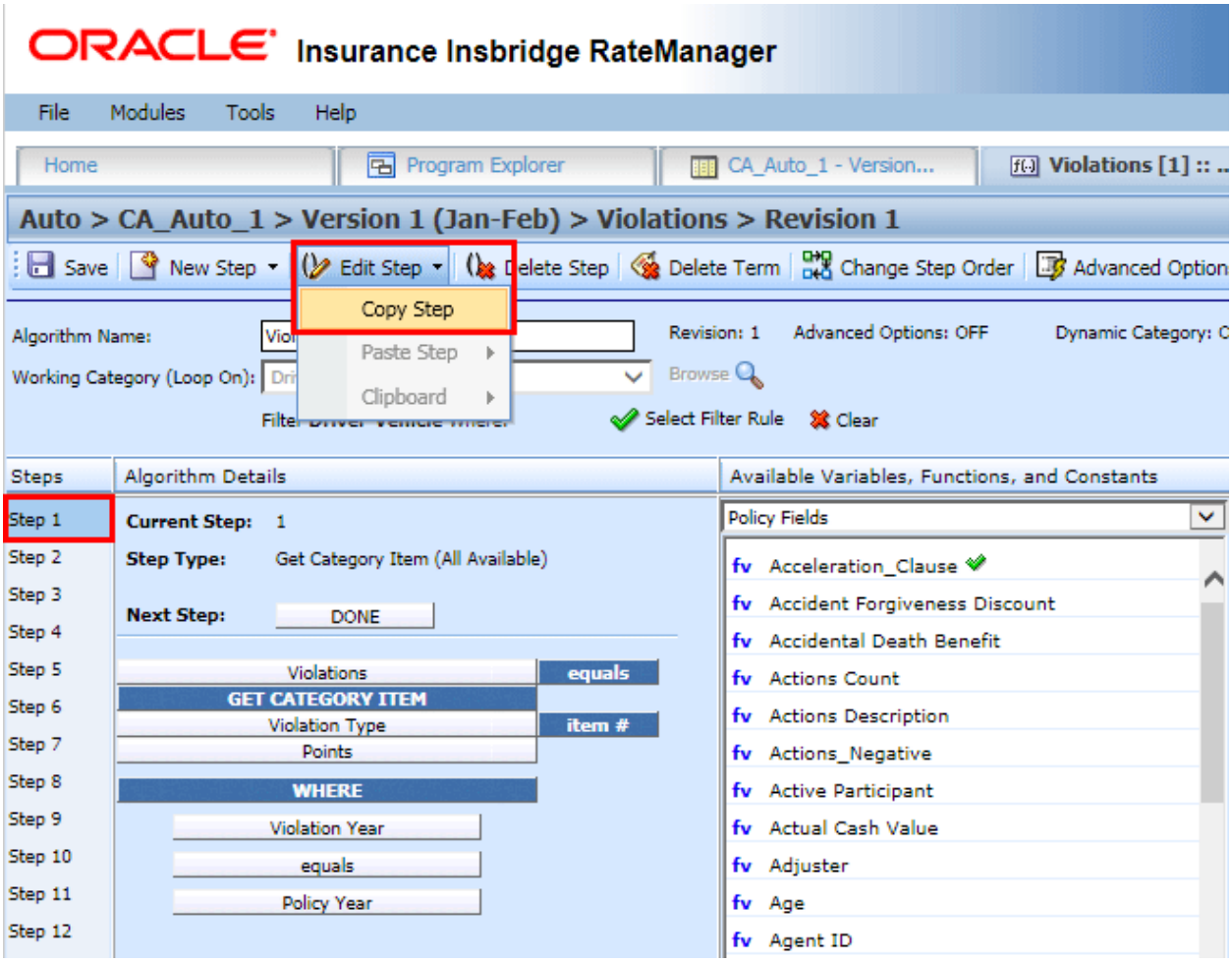


Figure 264 Copy Variable Menu

3. A confirmation window with the details of the step copied is displayed.

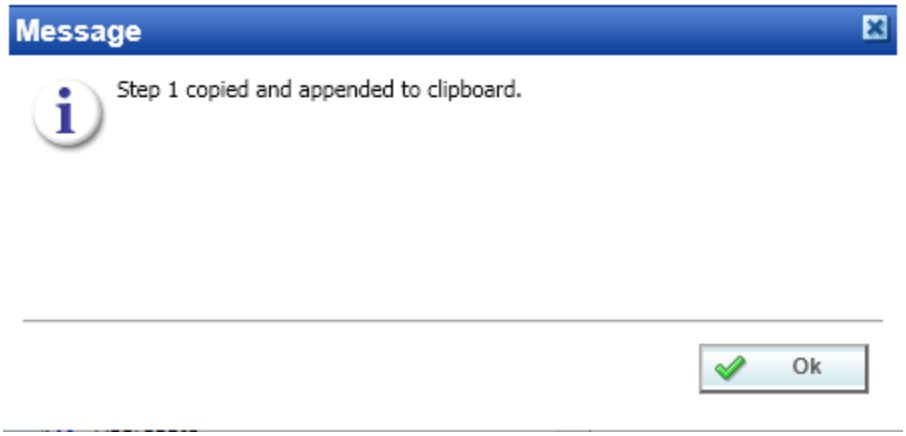


Figure 265 Successful Copy

4. If necessary, continue selecting steps.

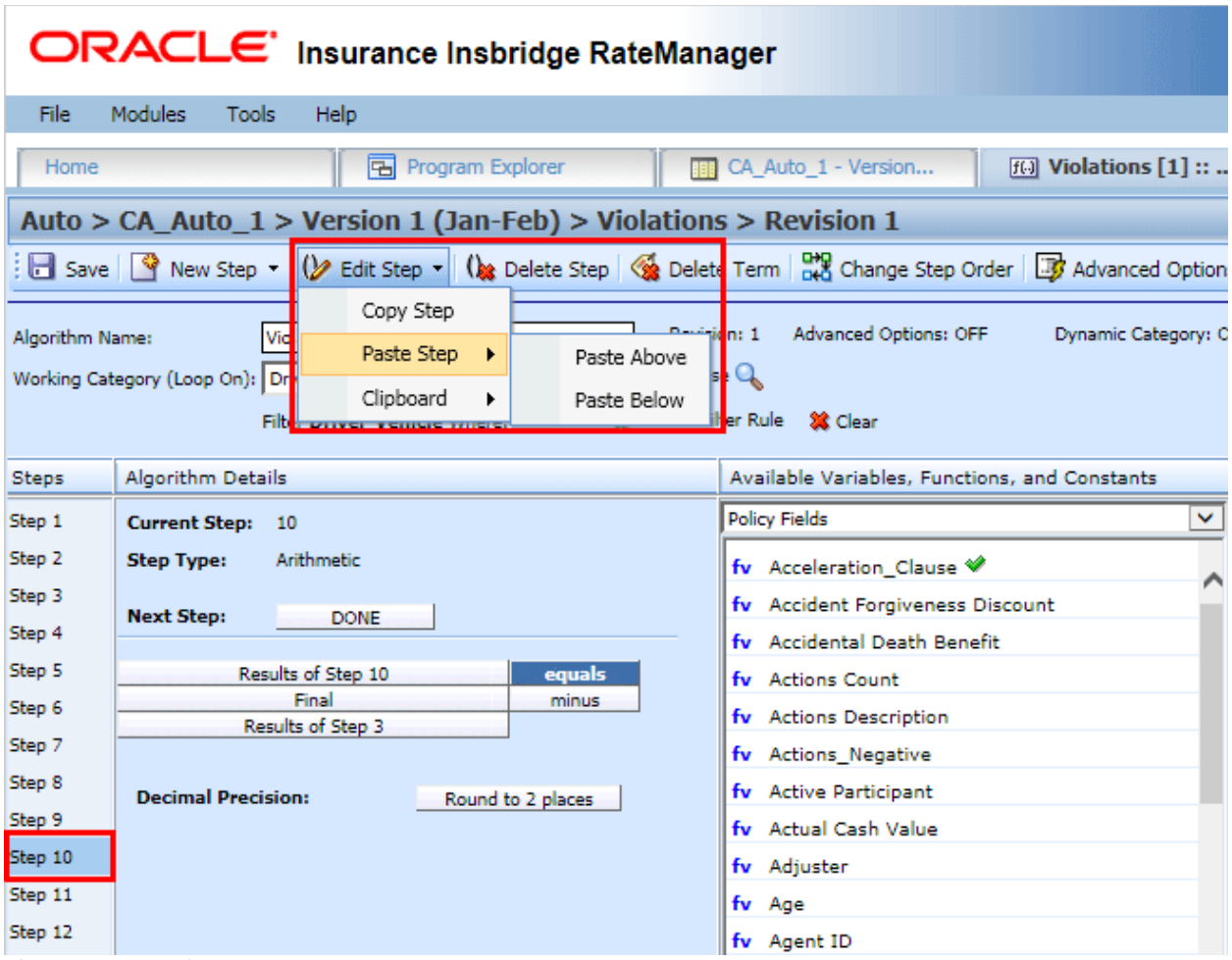


Figure 266 Pasting Steps

5. Select the step where you want to paste the copied steps.
6. Select **Edit Step**->**Paste Step** and select either past above or paste below.
7. A success message with the option to clear the clipboard is displayed.

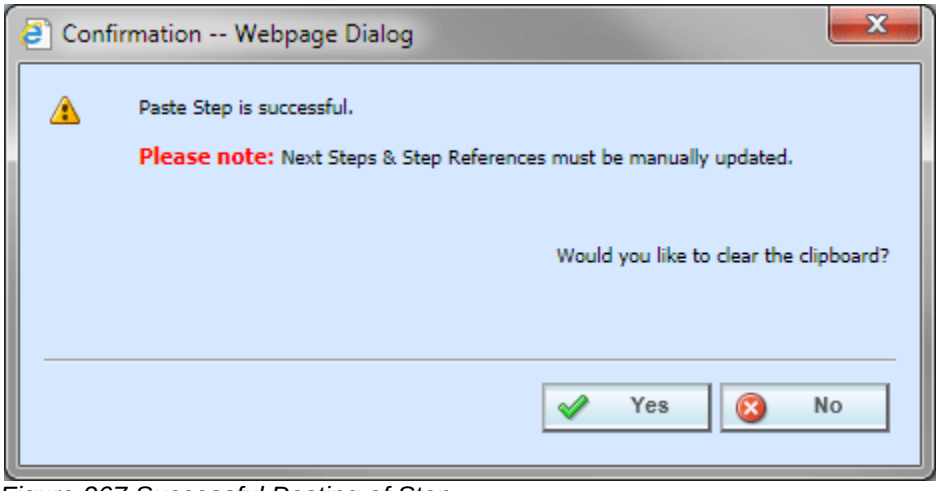


Figure 267 Successful Pasting of Step

8. Click Yes to clear the clipboard or No to keep the contents. Either choice closes the window.

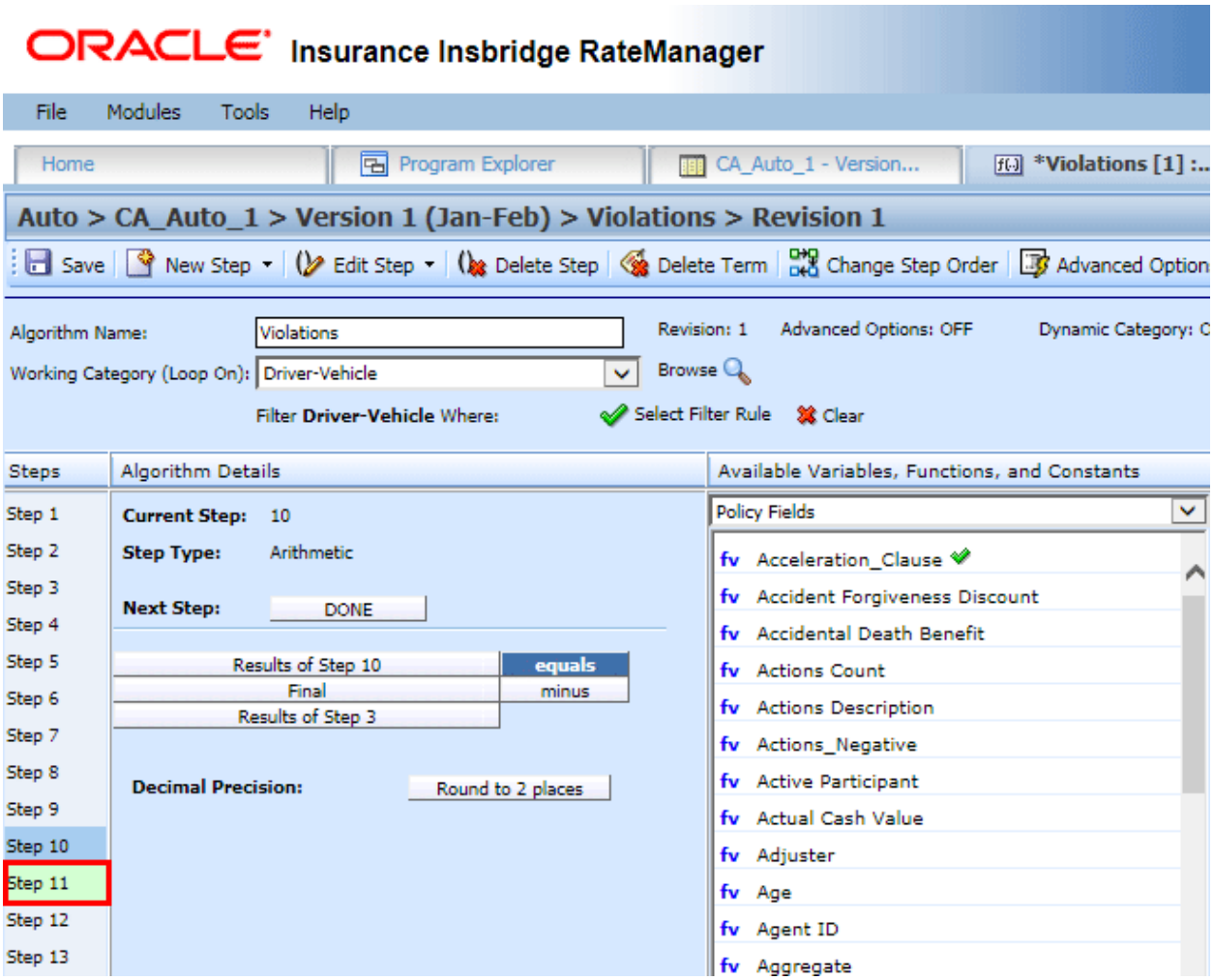


Figure 268 Pasted Step Highlighted

9. The pasted steps are highlighted either above or below the selected step. Reset any Next Step or Result of Step values. They are not updated automatically.

10. Save your work. Verify that the clipboard contents are no longer needed. When the element is closed, the clipboard is cleared.

NOTE: *Next Steps and Results of Steps are **NOT** updated when steps are added, moved or deleted. If steps are added, moved or deleted the values must be manually reset.*

Viewing and Clearing the Clipboard

The clipboard displays the currently copied steps in order. This option is available when there are contents on the clipboard. If the clipboard is empty, this option is grayed out.

The contents of the clipboard can be viewed by selecting Clipboard->View. Clipboard contents can only be viewed; individual steps cannot be deleted or moved.

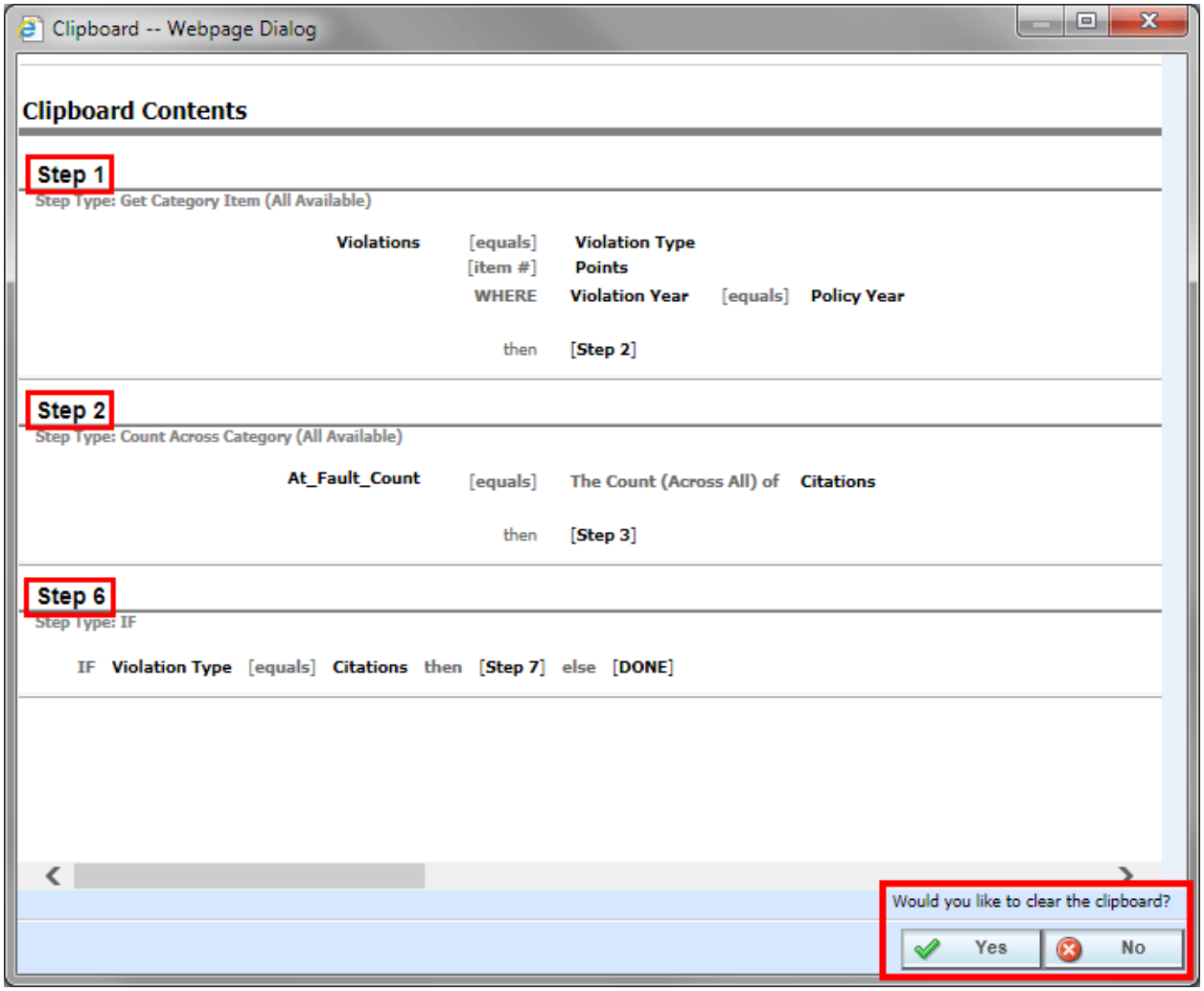


Figure 269 Clipboard Contents

Clipboard contents can be cleared after pasting. The clipboard can be cleared without pasting by selecting Clipboard->Clear. A success message is displayed after clearing. Clipboard contents are held for as long as the element is open and are unique to the element. Clipboards are cleared when the element is closed.

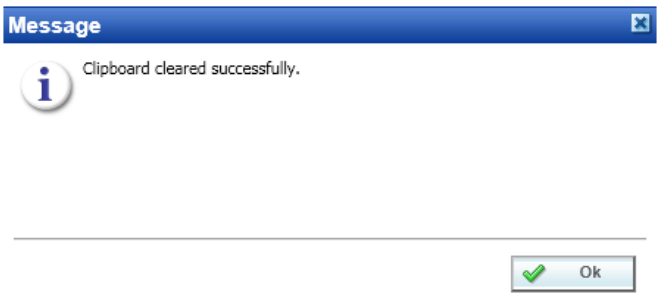


Figure 270 Successful Clearing of Clipboard

Setting the Next Step

For all steps, you have the ability to set the next step to be executed. On most step types, this is done by clicking the numbered step box next to **Next Step** and selecting the appropriate step from the menu. Next Step options are updated when the step is moved.

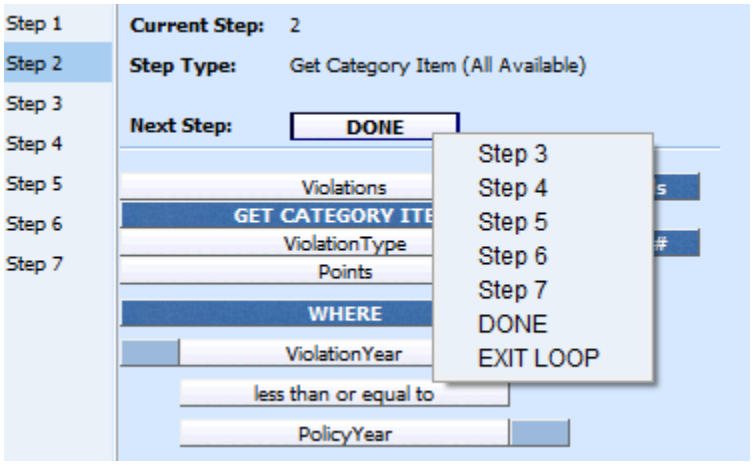


Figure 271 Setting the Next Step

The available choices for the Next Step are:

- **STEP #** – Selecting one of the steps listed sets that step as the next step to be processed. Only steps that come after the selected step are available. You cannot go backwards to a previous step or set the step to go to itself. The last step in the list does not have this option.
- **DONE** – Selecting **Done** exits the algorithm after the step is finished.
- **EXIT LOOP** – Selecting **Exit Loop** exits the algorithm after the step is finished and prevents any further iterations of the algorithm from executing.

For IF step types and Data Type steps, there are two **Next Step** boxes, labeled as **THEN** and **ELSE**. THEN is used to set the next step to execute if the statement evaluates to TRUE and ELSE is used to set the next step to execute if the statement evaluates to FALSE.

Setting the Next Step

To set the Next Step, click the Next Step field and select from the list.

Using Results of Step # Logic

Results of Step # are temporary holding fields that can be used to carry a value to a later step to be used as a term. Results of Step # cannot be carried out of the algorithm or calculated variable. Any values that need to be carried out should use a result variable. Results of Step # are used in two ways:

- **First, to establish the result of the step:** If you must carry the result of the step to another step down the list, you can select the use the result of current step to hold the value. This is the first

field in the step. In Figure 272 Results of Step, the first field in Step 10 is Results of Step 10. This value holds whatever result comes from the execution of this step.

- **Second, as a term in a step:** Once you have set a value in the Results of Step #, it can be used as a term in a step further down the steps list. In Figure 272 Results of Step, the last field in the step is Results of Step 3. The result from step 3 is being used in the execution of step 10.

Every step has Results of Step # available in the Step Variable listing. You can select to create a result for the current step or use the result from a previous step, one where you have specifically set the value. This means that you cannot select to use Results of Step 4 if you did not establish that Result of Step # in step 4. If you used a result variable or some other term, you will have to use that.

Since Results of Step # can be used in two ways, they are not validated for logic. Be aware that you cannot set the Results of Steps # to a step that is behind the current step. For example, if you are on step 10, you cannot establish the Result of Step 5. You can use the Results of Step 5 in the execution of step 10 but you cannot establish the result for a step other than the one where you are currently located.

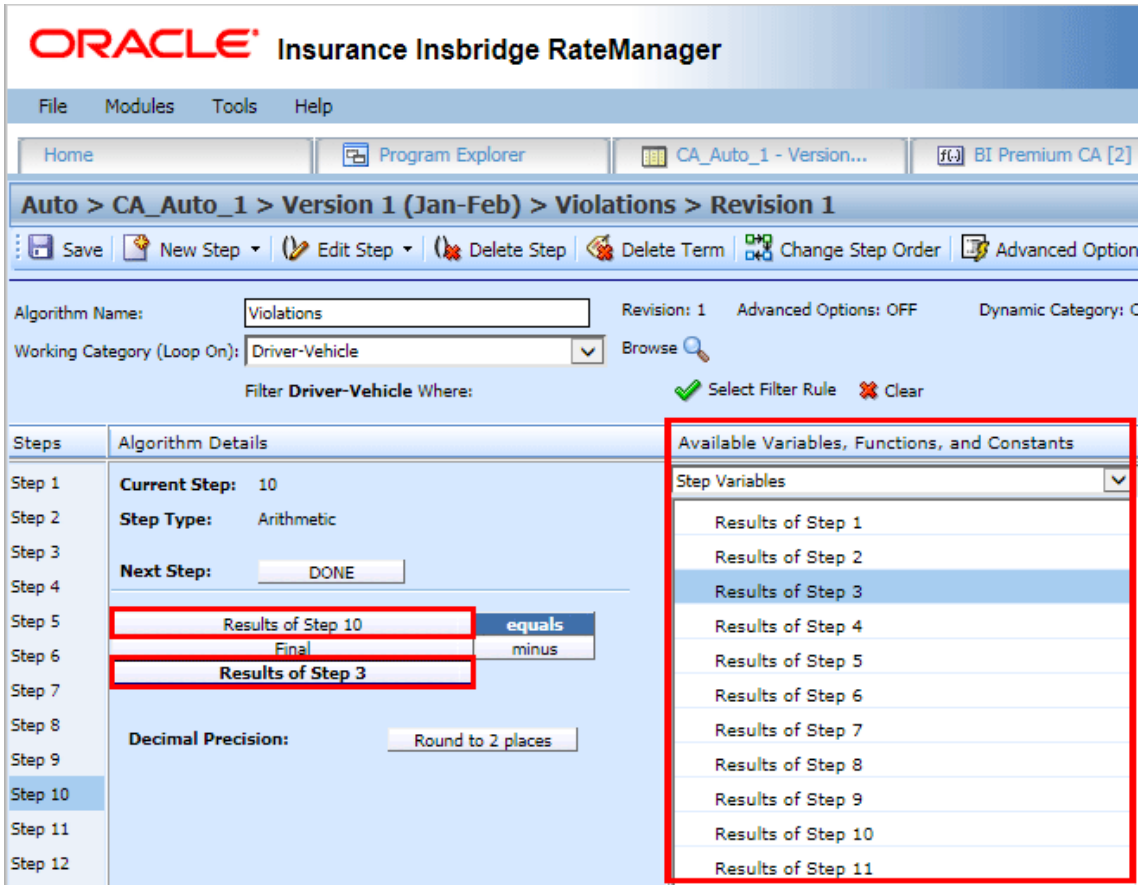


Figure 272 Results of Step

Results of Step are useful when calculating premiums where it is necessary to separate each calculation into individual steps. For example, the following sample algorithm, as it might be shown in a manual:

Premium = Base Rate x Territory Factor x Limit Factor x Credit A x Credit B
** Round to 0 places after each step*

In RateManager, you could break this algorithm into several arithmetic steps to properly round.

Within an algorithm or calculated variable, we can use a **Results of Step** term to incrementally save the end calculation from a step in order to use it in subsequent steps. The result from the step is only available in the current algorithm and will be lost once the algorithm completes processing.

For example: All the **Results of Step** terms are highlighted.

Step 1: *Results of Step 1 = Base Rate x Territory Factor*
Step 2: *Results of Step 2 = Results of Step 1 x Limit Factor*
Step 3: *Results of Step 3 = Results of Step 2 x Credit A*
Step 4: *PremiumX = Results of Step 3 x Credit B*

The final step would contain the result variable (PremiumX) we want to carry out of the algorithm. The results of steps cannot be carried out of the algorithm.

NOTE: *Next Step and Results of Step # are **NOT** updated when steps are added, moved or deleted. If steps are added, moved or deleted, you must reset the Next Step and Result of Step # values.*

Updating Terms

To complete or update terms within the step, highlight the term that you want to complete or update in the Algorithm Details area. Then click the variable you need from the Available Variable, Functions, and Constants. The variable is populated in the term.

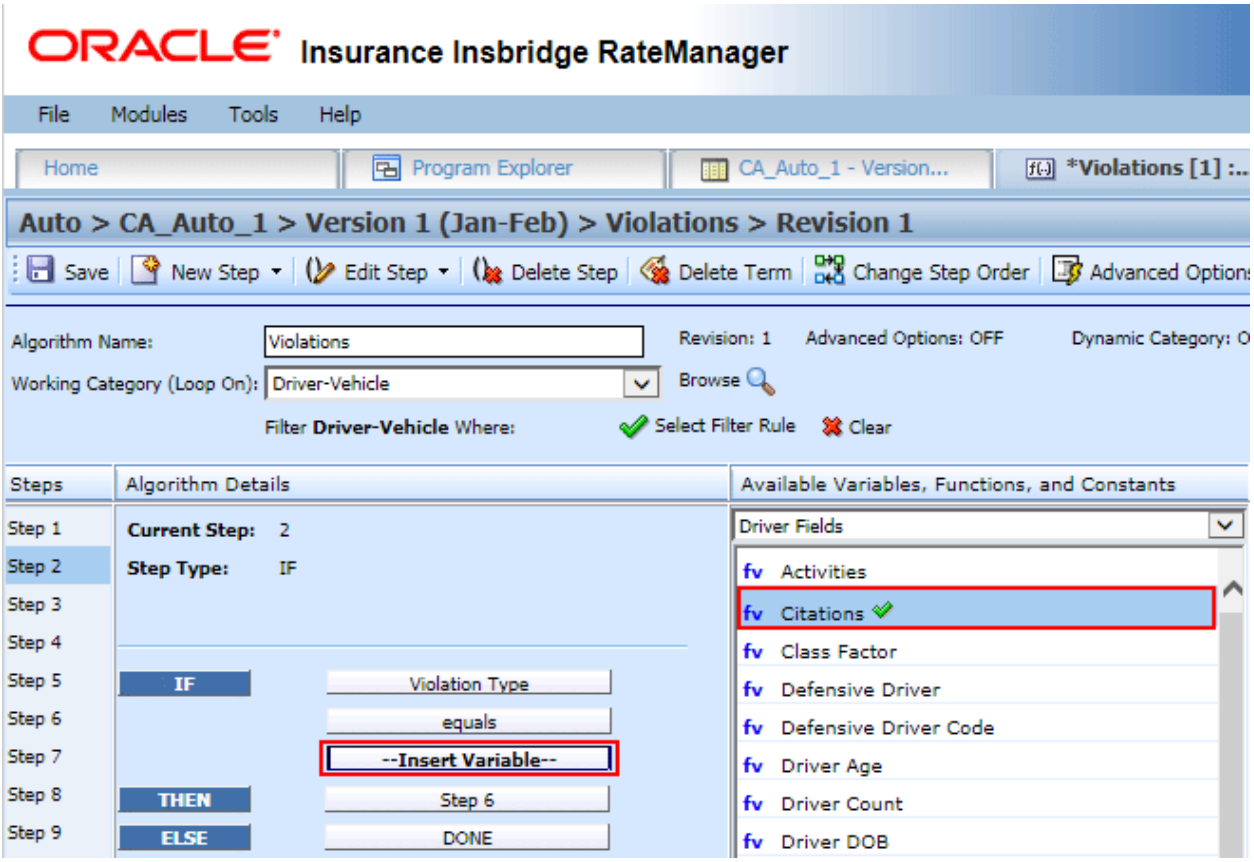


Figure 273 Editing Steps

NOTE: *Steps are not validated for content. If you do not select a value for a term no error message is displayed.*

Right Click Menus

Most terms have a right click menu.

- **View Report:** Brings an information screen containing XML field ID, type and category.
- **View Where Used:** Brings up a View Where Used report.
- **Custom Value:** Enter in a custom value. Not available on every step.
- **Delete:** Available on Arithmetic and String Addition steps only. Removes a term in the step.

Comparison Operators

Comparison operators are used in IF steps, arithmetic steps, and data functions. Arithmetic comparisons only can be performed on numbers (integers and decimals). RateManager limits your options when comparing strings to Equals and Not Equals. An operator may be used to specify a range of numbers by setting the first criteria as Greater Than and the second criteria as Less Than, or to specify an exact match by setting the operator to Equal. Arithmetic steps can also use bitwise OR and bitwise AND. See the Arithmetic step for details.

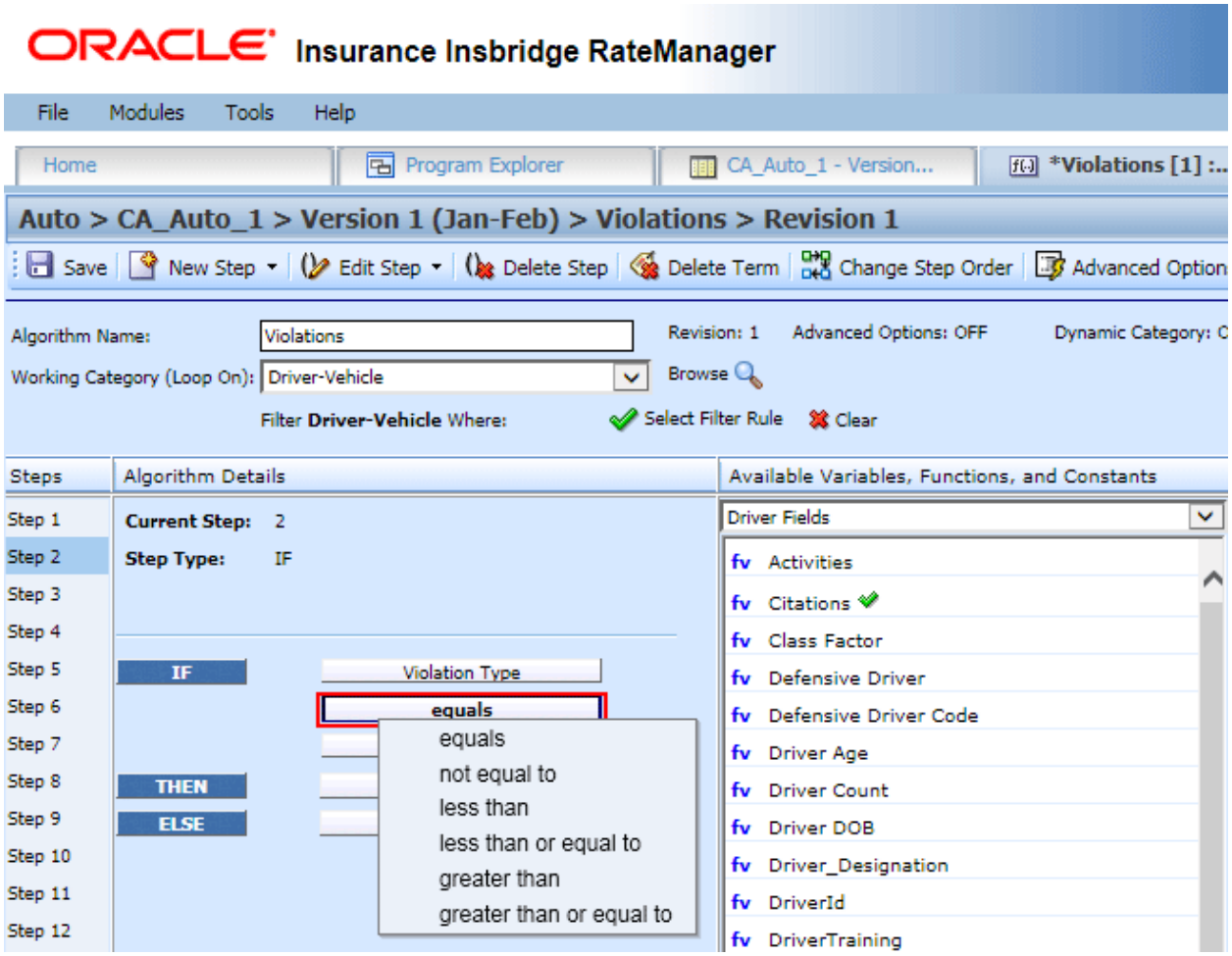


Figure 274 Setting the Step Operator

The available comparison operators are:

=	Equals
<>	Not Equals
<	Less Than
<=	Less Than Or Equal To
>	Greater Than
>=	Greater Than Or Equal To

Figure 275 Comparison Operators Table

Setting Rounding

To set the rounding to be used at the end of the step, click the box next to **Decimal Precision** and select the appropriate rounding from the menu. Rounding can be to 10 decimal places.

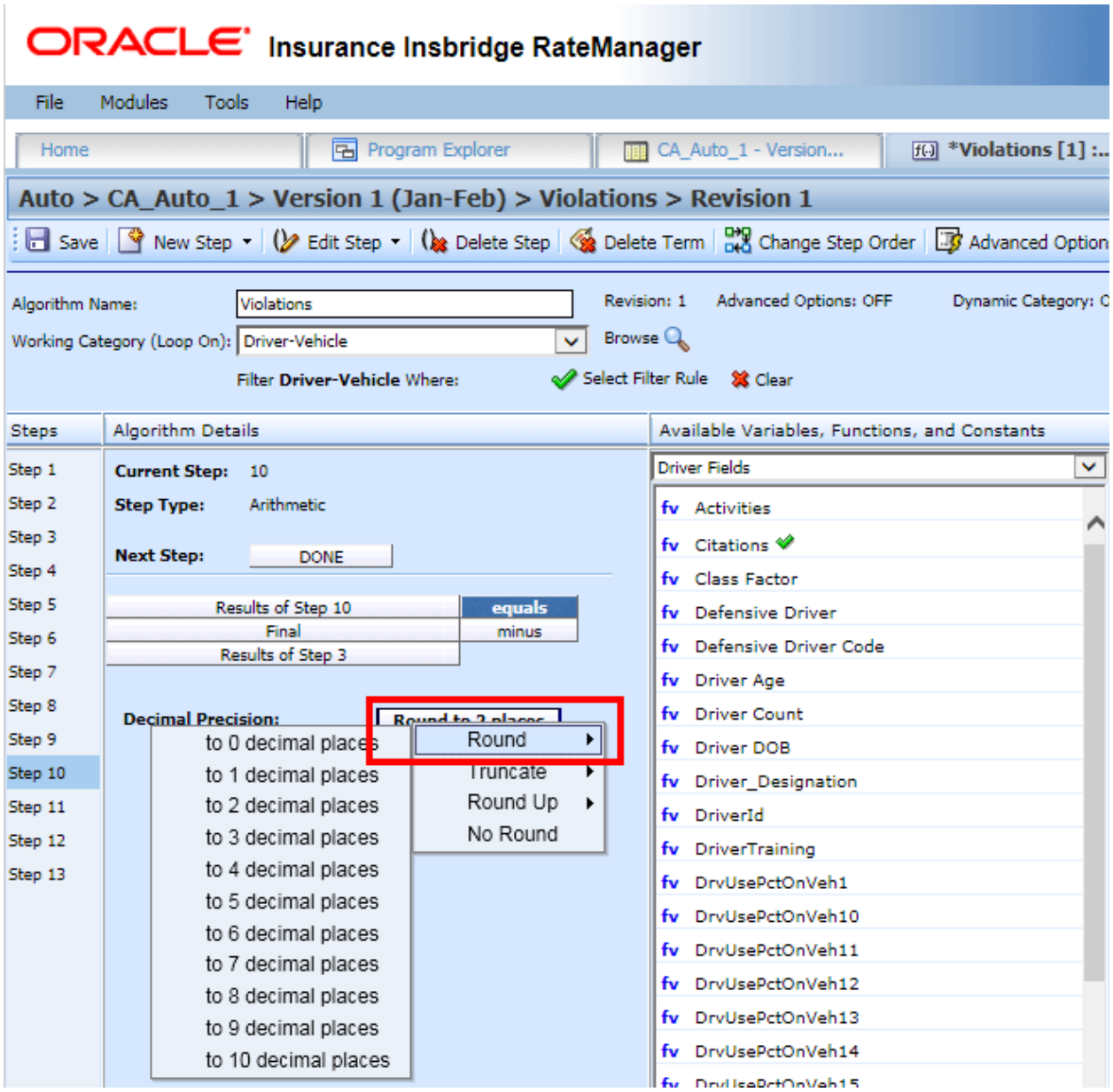


Figure 276 Setting the Rounding Option

Decimal Precision

Many math functions and trig functions allow you to set the rounding parameters at the end of the step.

- **Round:** Rounds the number using standard rounding rules. Rounding can be set anywhere between 0 and 10 digits.
- **Truncate:** All or a portion of the decimal are dropped. Can be set between 0 and 10 digits.
- **Round Up:** Will round the number, or a portion of it, to the next higher number. Can be set between 0 and 10 digits.
- **No Round:** No rounding occurs on the step.

In RateManager, rounding is always performed Round Half Up. Halfway values, .5 and over, are rounded up. .4 and lower are rounded down.

WHERE CLAUSE

The Where clause specifies the criteria used when executing select steps. By using specific criteria, you can further define when the step is to be executed. For example:

- A **Get Category Item** step can be used to get the 2nd violation for each driver, but no other criteria can be considered.

Driver Second Violation Type	[equals]	Type
	[item #]	2

Adding a Where clause runs this step with a specified condition such as, only count the violation occurrence if the points value is greater than 1.

Driver Second Violation Type	[equals]	Type
	[item #]	2
	[where]	
Driver Violation **Points**	[greater than]	1

- If we have 5 violations under Driver1 (v1, v2, v3, v4, v5), a **Get Category Item Use Current Path** also could be used to get the 2nd category instance, in this case violations.

Category Item	[equals]	Instance
	[item #]	2

Adding a Where clause runs this step to obtain the second instance in the current category path with a specified condition such as, only count if the points value is greater than 1.

Category Item	[equals]	Instance
	[item #]	2
	[where]	
Points	[greater than]	1

When the step is processed, if only v1, v3, and v5 get passed into the step, the where condition could still be satisfied. The “v3” value would be the second instance of the category and it would meet the definition of the where clause.

- A **Sum Across** step can be used to determine the total premium for multiple instances of a particular scheduled property type.

Scheduled Jewelry Premium	[equals]	The Sum (Across All) of Scheduled Item Premium

Adding a Where clause to define a specific criteria would more clearly define the results returned.

Scheduled Jewelry Premium	[equals]	The Sum (Across All) of Scheduled Item Premium
	[where]	
Scheduled Item **Type**	[equals]	Jewelry

The Where clause is available on select step types.

- **Category Items:** These steps are used to get or set a specific instance of a category.
  - Get Category Item (Use Current Path)
  - Set Category Item (Use Current Path)
  - Get Category Item (All Available)
  - Set Category Item (All Available)
  - Get Ranked Category Item
  - Set Ranked Category Item

- **Count Across:** These steps are used to determine how many instances of a specific category exist.
  - Count Across Category (All Available)
  - Count Across Category (Use Current Path)

- **Sum Across:** These steps are used to determine the sum of all instances of a specific category.
  - Sum across Category (All Available)
  - Sum across Category (Use Current Path)

- **Ranking:** These steps rank all instances of a specified category, either highest to lowest or lowest to highest.
  - Rank across Category (All Available) (High-to-Low)
  - Rank across Category (All Available) (Low-to-High)
  - Rank across Category (High-to-Low)
  - Rank across Category (Low-to-High)

The Where clause can be further defined by using OR and AND conditions. Each eligible step can contain only one Where clause. Not all steps will allow for a Where clause to be created.

Where Clause Categories

The where clause is structured as an expression with two terms and operator. For example,

Where (Term 1) = (Term 2)

When creating a where clause, Term 1 should be from the variables instance category tree. Term 2 should be from the algorithm or calculated variable that is being executed.

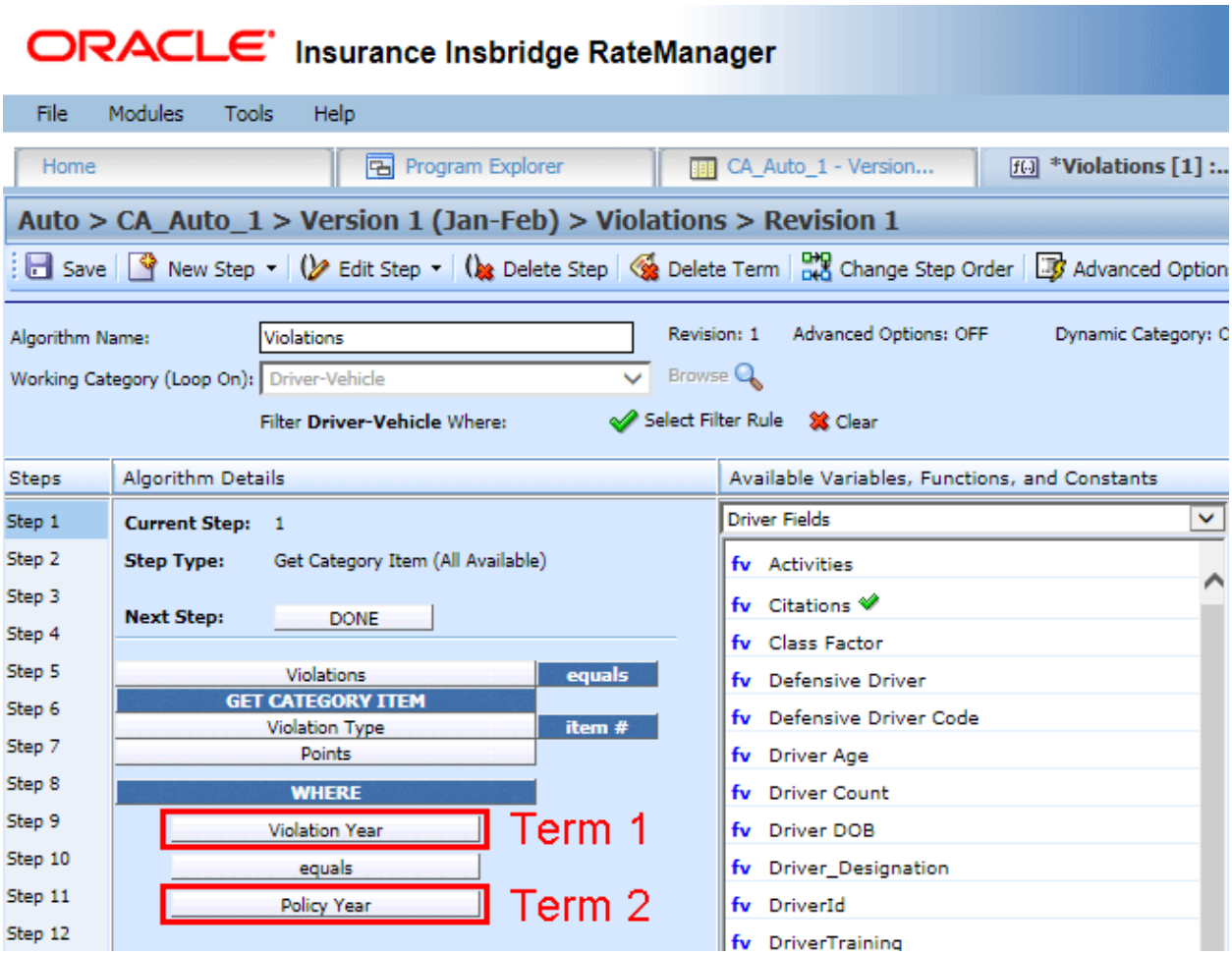


Figure 277 Where Clause Category Structure

CONDITIONS

Conditions can be added to select step types and to Do While Looping on the Algorithm Advanced Option. Conditions allow for parameters to be set to filter or exclude data that falls outside the scope of data values needed. Conditions are defined in statements, for example:

- Get a value(*x*) only when A equals B and B is greater than C.

  GET Value (*x*)

	[where]	
A	[equals]	B
	[AND]	
B	[greater than]	C

- IF A equals B and B is greater than C then proceed if not then stop.

IF: A	[equals]	B

	[AND]	
B	[greater than]	C
	[Then]	
NEXT STEP		
	[Else]	
Done		

There are two Conditions that can be used in select step types:

- **OR** Condition where at least one statement is true.
- **AND** Condition where all statements are true.

Category Items, Count Across, Sum Across, and Ranking can be built using the Where clause, OR, and AND conditions.

Multiple and complex IF Statements are built using OR and AND conditions. Simple IF Statements do not have conditions. The Where clause is not available in IF Statements.

OR Condition

Use the OR condition when you need a statement that has to contain a minimum of one true value. An IF statement with only OR conditions will be run until one true value is found. When the true value is met, the THEN segment will be processed. If there are no true values, the ELSE segment will be processed.

- **For example:** If you wanted to run a statement where Variable A is greater than zero, OR Variable B is less than 1000, OR Variable C [=] 100, then continue, else stop.

IF: Variable A	[greater than]	0
	[OR]	
Variable B	[less than]	1000
	[OR]	
Variable C	[equals]	100

(If **any** of these conditions are **true**) THEN Step 2

(If **none** of these conditions are **true**) ELSE Done

AND Condition

Use the AND condition when you need a statement that has to contain only true values. An IF Statement with only AND conditions will be run until all segments of the statement have been validated. If all values are returned true, the THEN segment will be processed. If there are any values that are not true, the ELSE segment will be processed.

- **For example:** If you wanted to run a statement where Variable D is greater than zero, AND Variable E is less than 1000, AND Variable F [=] 100, then continue, else stop.

IF: Variable D	[greater than]	0

	[AND]	
Variable E	[less than]	1000
	[AND]	
Variable F	[equals]	100

(If **all** of these conditions are **True**) THEN Step 3

(If **any** of these conditions are **False**) ELSE Done

Parentheses

Parentheses can be added to a complex IF (Normal) statements, Where clauses, and Do While Looping on the Algorithm Advanced Option to group conditions in a specific order and to make the statement more precise. Conditions that are grouped in parentheses are processed first starting with the inner most group and working through the statement. After grouped conditions have been processed the RateManager order of operations, where the first equation is processed and then the next equation and so on, will be used.

For example: If you wanted to execute a statement where Variable A is greater than zero or Variable B is less than 1000, OR Variable C [=] 100 and Variable D is greater than zero, then continue, else stop.

You would group the first two statements in parentheses and the next two statements in parentheses.

Order	Condition	Element
Group 1		**(**IF Variable A [>] 0 OR IF Variable B [<] 1000**)**
	OR	
Group 2		**(**IF Variable C [=] 100 AND IF Variable D [>] 0**)**

IF: (Variable A	[greater than]	0
	[OR]	
Variable B	[less than]	1000 **)**
	[OR]	
(Variable C	[equals]	100
	[AND]	
Variable D	[greater than]	0 **)**

This statement is validating:

- If Group 1 or Group 2 returns a true value, then the entire statement is evaluated to be true and the Then segment is processed.

- If Group 1 and Group 2 both fail, the entire statement evaluates to false and the Else segment is processed.

Nested Parentheses

Parentheses also can be nested with the inner most grouping processed first then moving outward. Opening parenthesis is available in the field in front of the variable and closing parenthesis is available in the field at the end of the variable.

Parentheses must be in pairs. If there is an open without a close parenthesis or a close without an open, an error message will be displayed. Parentheses are validated for pairing. No validation is done to verify the content of the parentheses.

For example: Expanding the parentheses example and add another qualifier to each grouping. Execute a statement where Variable A is greater than zero or Variable B is less than 1000, *AND Variable E is less than 1000*, OR Variable C [=] 100 and Variable D is greater than zero, *AND Variable F [=] 100*, then continue, else stop.

You would still group the first two statements in parentheses. Next you add a set of parentheses to include the additional qualifying statement and group 1. All three statements would be group one with first two statements being validated first, then the final qualifying statement.

The next two statements would also be grouped in parentheses and another set of parentheses would be added to include the additional qualifier for that statement. All three statements would be group two with first two statements being validated first, then the final qualifying statement.

Order	Condition	Element
(Group 1		**(**IF Variable A [>] 0 OR IF Variable B [<] 1000**)**
Qualifier 1**)**	AND	IF Variable E [<] 1000
	OR	
(Group 2		**(**IF Variable C [=] 100 OR IF Variable D [>] 0**)**
Qualifier 2**)**	AND	IF Variable F [=] 100

IF: ((Variable A	[greater than]	0
	[OR]	
Variable B	[less than]	1000 **)**
	[AND]	
Variable E	[less than]	1000 **)**
	[OR]	
((Variable C	[equals]	100

	[OR]	
Variable D	[greater than]	0 **)**
	[AND]	
Variable F	[equals]	100 **)**

This statement is validating:

- If Group 1 or Group 2 returns a true value, then the entire statement is evaluated to be true and the Then segment is processed.
- If Group 1 or Group 2 both fail, the entire statement is evaluated to false and the Else segment is processed.

To add parentheses, select the opening parenthesis then select the closing parenthesis. If grouping parentheses are to be nested, begin with the inner most pair and work outwards.

Viewing Parentheses in the Algorithm Report

Parentheses are visible in the Algorithm Details section on the condition where they were added. Parentheses pairings are color coded in the Algorithm Report section.

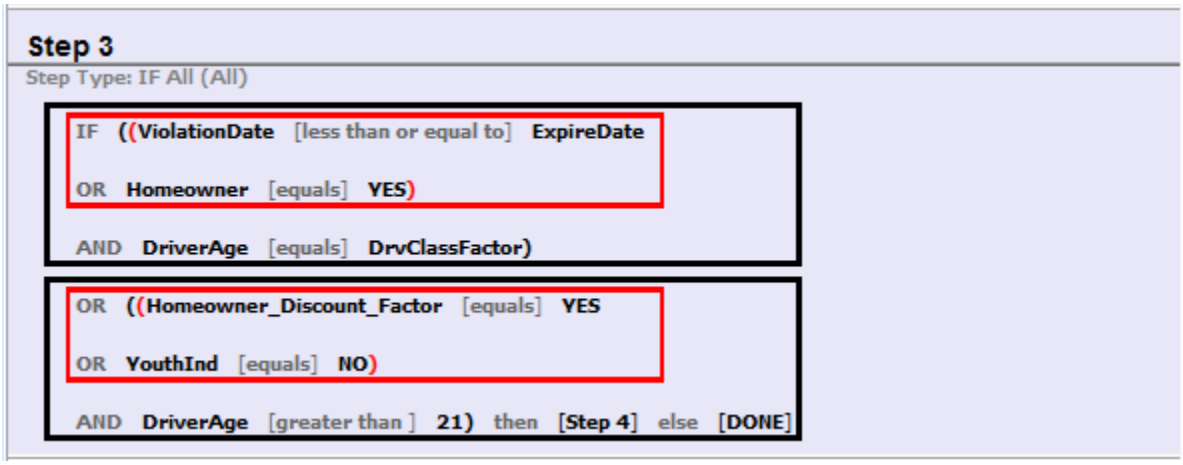


Figure 278 Parentheses in The Algorithm Report

Parentheses colors determine the level of the group. There are six colors that can differentiate the groupings. If there are more than six groupings the color will duplicate.

- **Black** is the outermost level
- **Red**
- **Purple**
- **Green**
- **Brown**
- **Blue** is the inner most level

The processing begins with the inner most level and works outwards. You can have multiple levels of inner most parentheses and next level and so on. Depending upon your configuration, you may see multiple pairs of blue parentheses or multiple pairs of purple parentheses and so on. The first parenthesis will be paired up with the first matching end parenthesis.

Adding a Where Clause

1. Select the step where you want to add the Where clause.

2. Click the step type name to display the Add Where clause. This will not be a variable field but a solid color field with the step name.

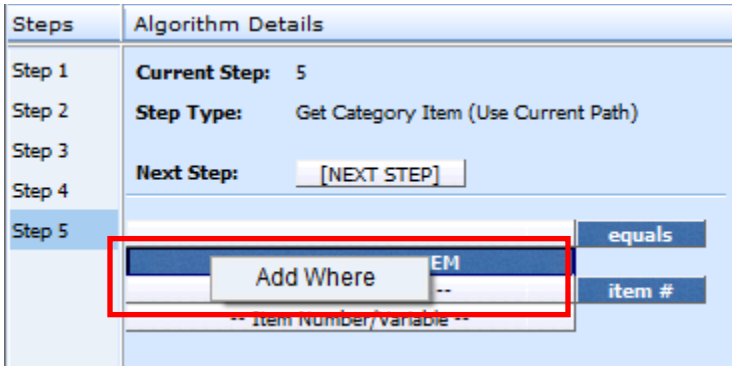


Figure 279 Adding a Where Clause

3. Click the Add Where option to add a Where Clause under the step.

4. Enter in the variables required and select the operator needed.

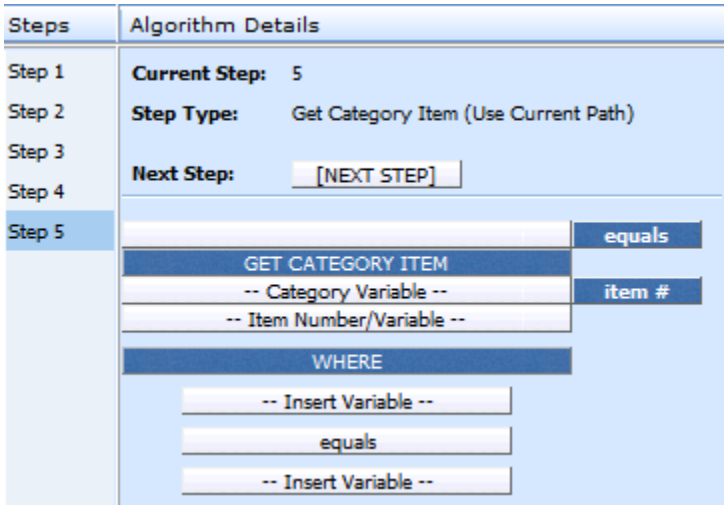


Figure 280 Completing the Where Clause

5. When complete, save your work.

Editing a Where Clause

The terms of a Where clause can be edited at any time. Terms are edited the same way as a step term. Please see Updating Terms.

Deleting a Where Clause

Where clauses can be deleted at any time. Deleting a Where clause deletes all conditions in the clause as well. No warning is given. When you click Delete, the entire clause is removed immediately.

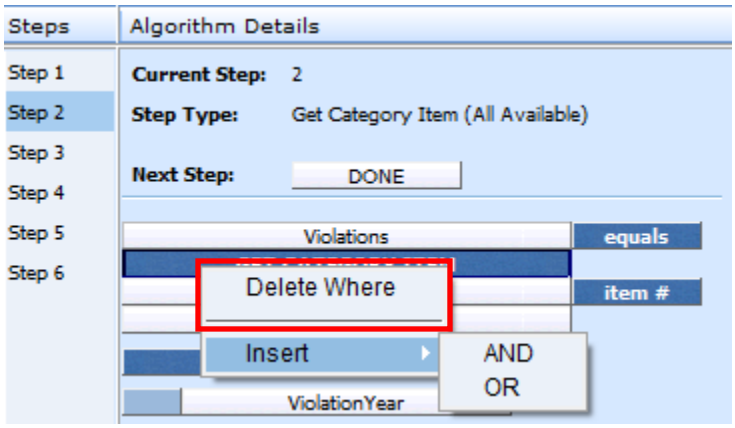


Figure 281 Deleting a Where Clause

To delete a clause, click the Where name field and select Delete Where. The clause and any condtitions will be removed.

Adding Conditions

Conditions can be added by clicking the Where name field and selecting Insert and the option you want from the menu, AND or OR.

The Where condition can be removed by clicking the Where name field and selecting Delte Where from the menu.

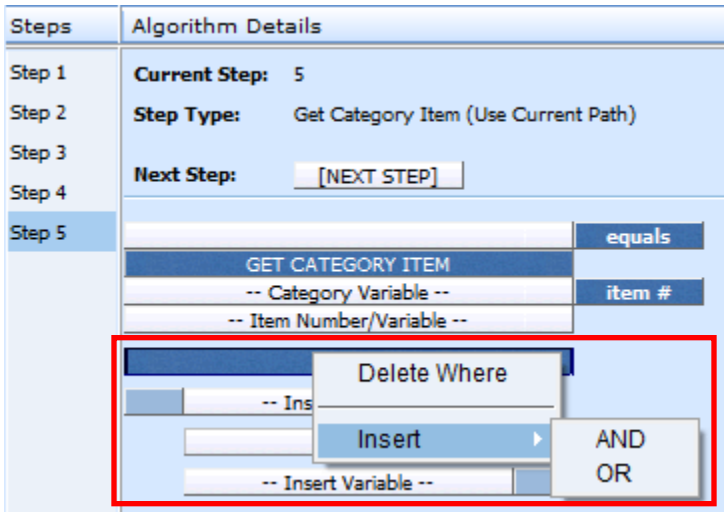


Figure 282 Adding Conditions to a Where Clause

You also can click the step type name to view the menu options. Options are:

- **Delete Where:** Deletes the entire Where clause, including and conditions. No warning is given. The action is immediate.

- **Insert AND:** Inserts and AND condition under the Where clause.
- **Insert OR:** Inserts an OR condition under the Where clause.

Editing Conditions

You can change the condition, insert additional conditions or delete a condition statement by clicking the condition (AND or OR) blue box and selecting from the menu.

Options are:

- **AND:** The AND in the upper portion of the menu changes the selected condition to an AND type condition.
- **OR:** The OR in the upper portion of the menu changes the selected condition to an OR type condition.
- **Insert AND:** Inserts an AND condition under the selected condition.
- **Insert OR:** Inserts an OR condition under the selected condition.
- **Delete:** Deletes the selected condition. No warning is given. The action is immediate.

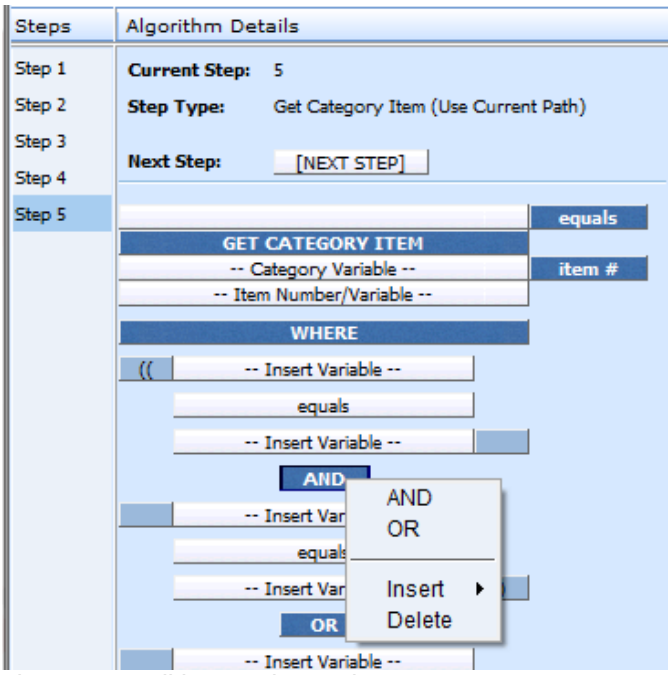


Figure 283 Editing a Where Clause

Deleting Conditions

You can delete a condition you do not need. Right click the AND or the OR blue box of the condition you want to delete. A menu will be displayed. Click **Delete**. The screen will refresh and the condition is removed.

Conditions are deleted one at a time.

Adding Parentheses

Parentheses can be added to IF (Normal) statement conditions, Where clauses, and Do While Looping on the Algorithm Advanced Option to group conditions in a specific order and to make the statement more precise. It is recommended to add parentheses after all conditions have been entered.

Parentheses can be added only to algorithms that have AND and OR conditions. Any algorithm that has all AND or all OR conditions will not have parentheses options.

Only IF (Normal) statements can have parentheses added. Other IF type statements do not have the option to add parentheses.

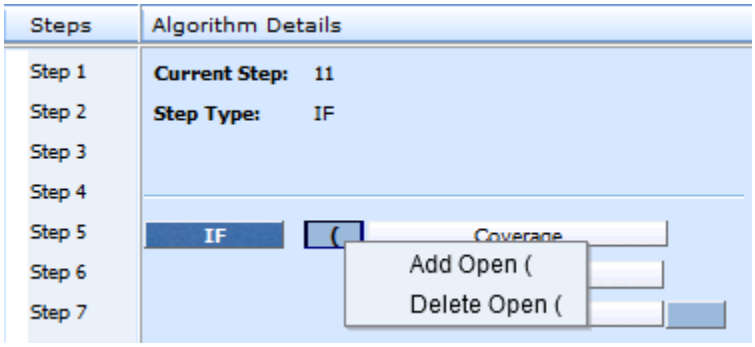


Figure 284 Adding an Open Parentheses

1. Select the first condition or term you want enclosed in parentheses. Click the field in front of the term. If no other parentheses have been entered, the only option will be Add Open (.

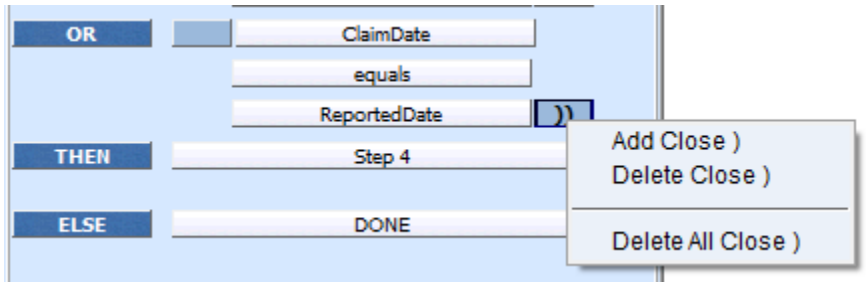


Figure 285 Adding a Close Parentheses

2. Select the last condition or term you want enclosed in parentheses. Click the field at the end of the term. If no other parentheses have been entered, the only option will be Add Close).

3. Save your work. The system will validate for pairs but not for logic. If you receive an error, you will have to correct before you can save.

You can have more than one open or close parentheses in a field. Multiple open or close parentheses are used in nested groupings. To validate the pairings, look in the Algorithm Report area on the tab. Parentheses groupings will be color coded.

Editing Parentheses

The options to edit depend on what you have entered. The options are:

- **Add Open (:** Adds a single open parentheses.
- **Delete Open (:** Removes a single open parentheses.
- **Delete All Open (:** Removes all open parentheses in the field. If you have other open parentheses in other fields, you will have to remove them in the same way.
- **Add Close):** Adds a single close parentheses.
- **Delete Close):** Removes a single close parentheses.
- **Delete All Close):** Removes all close parentheses in the field. If you have other close parentheses in other fields, you will have to remove them in the same way.

Deleting Parentheses

Parentheses can be deleted at any time. No warning is given. When you select to delete, the parentheses will be removed immediately.

You can review the parentheses pairings in the Algorithm Report area of the tab.

Moving Steps

You can move a step at any time to any position. Steps are moved one at a time. There are two ways to move steps.

Drop and Drag

Drop and drag allow you to highlight the step you want to move and drag it to the location you want. The current position is in the step details. Where you have the step positioned to move is in the highlighted box. A line is positioned in the steps column showing where the step will be dropped.

Figure 286 Moving Steps

1. Highlight the step you want to move in the **Steps** column.
2. Drag it to a new location in the list and drop it.

Using the Change Step Order Button

The Change Step Order button may be easier to use if you have a lot of steps. You can select the exact location you want from a list of steps.

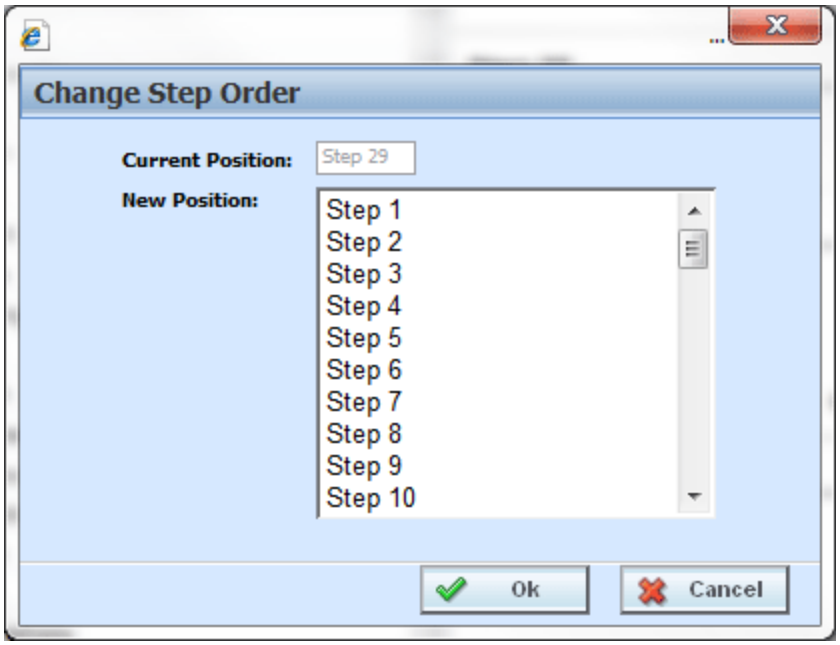


Figure 287 Change Step Order

1. To use the Change Step Order button, highlight the step you want to move in the **Steps** column.
2. Click the **Change Step Order** button. A window is displayed.
3. Select the new position where you want the step.
4. Click **Ok**. The window closes and the step is in the new location.

Deleting Term

The Arithmetic step and the String Addition step can have terms in the step deleted. The last term cannot be deleted. There must be at least one term. This function is only available to Arithmetic and String Addition steps. Other steps types have a set number of fields that cannot be deleted. The fields can be updated.

1. To delete a term, select the term in the **Step**.

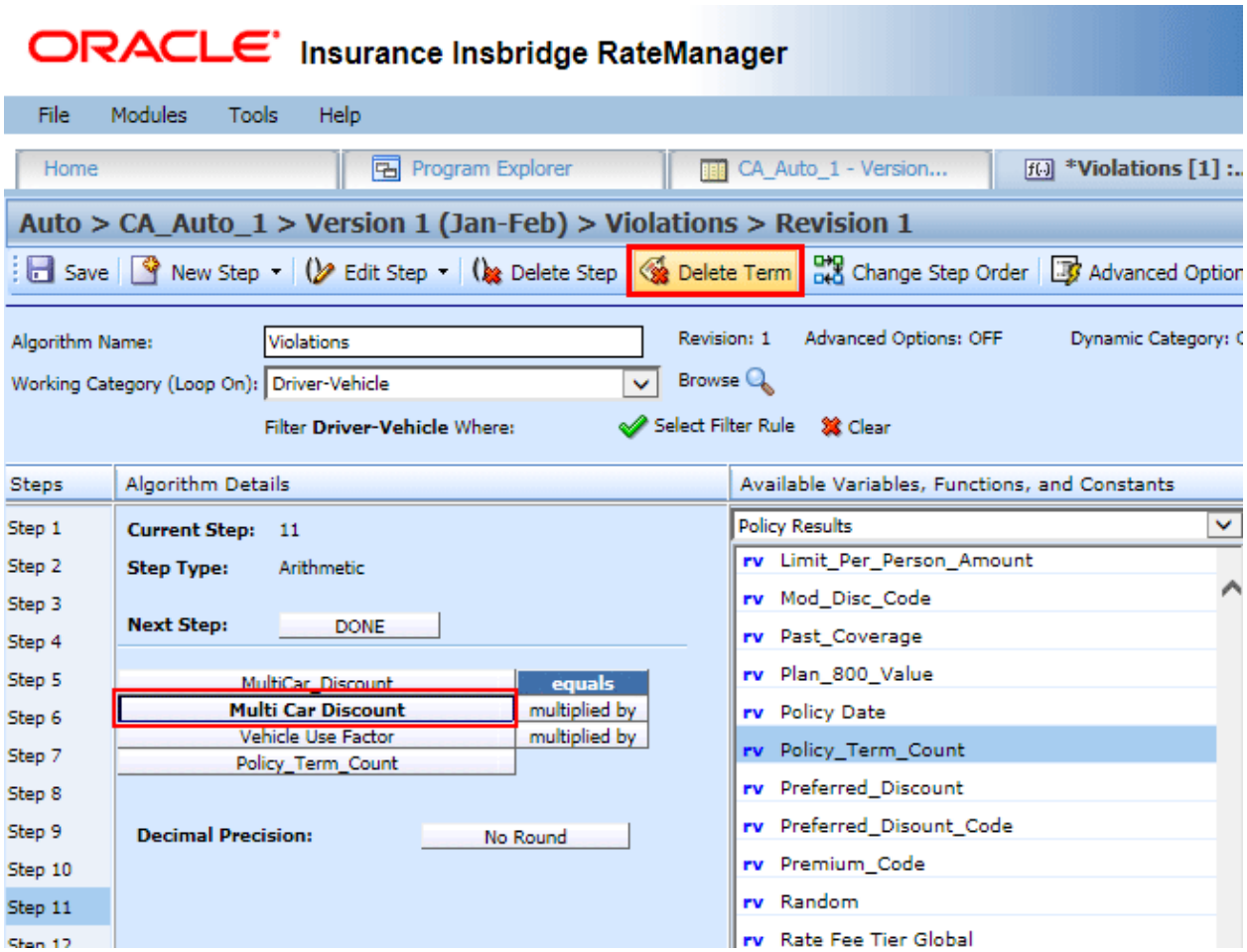


Figure 288 Deleting a Step Term

2. Click **Delete Term** in the top bar menu.

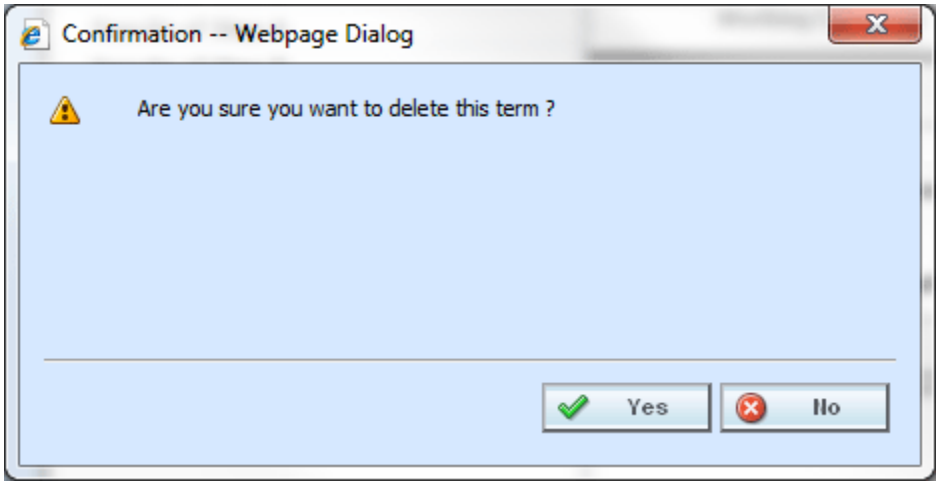


Figure 289 Delete Step Term Warning

3. You will be asked to confirm your deletion. Select **Yes** to delete and **No** to return to the tab.

Deleting Steps

Steps can be deleted at any time. The last step cannot be deleted. Algorithms, calculated variables, drivers assignments, and filter rules must contain at least one step.

1. To delete a step, select the step in the **Steps** column.

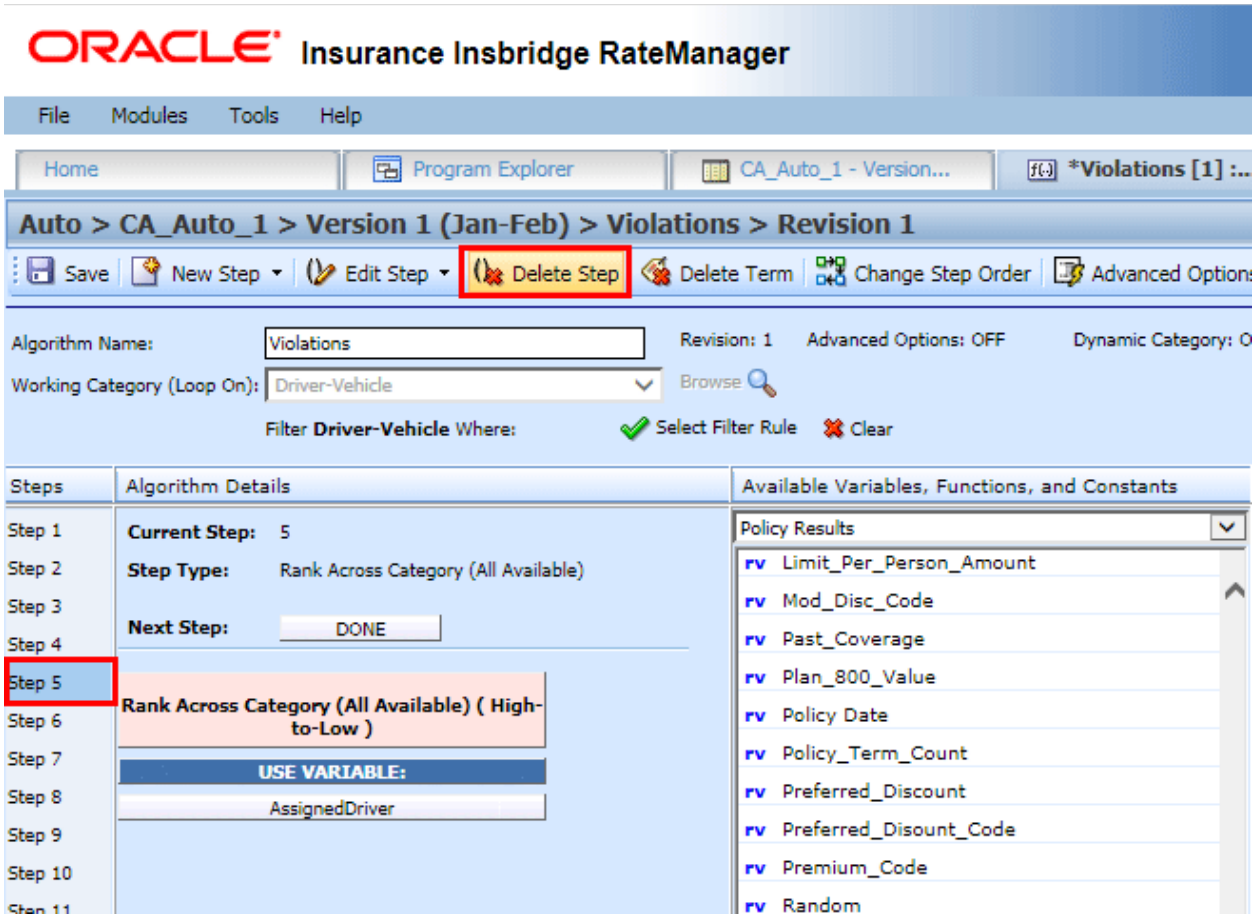


Figure 290 Deleting a Step

2. Click **Delete Step** in the top bar menu.

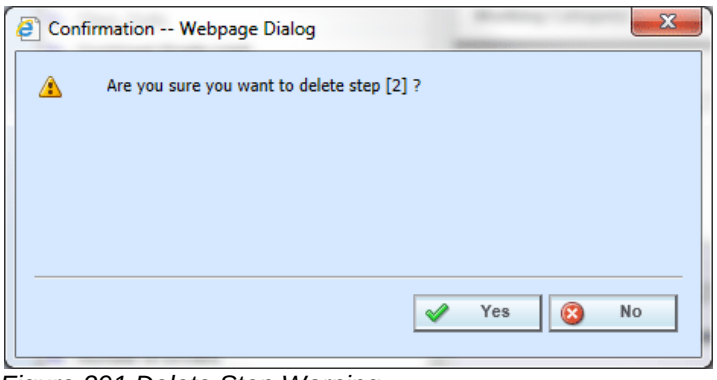


Figure 291 Delete Step Warning

3. You are asked to confirm your deletion. Select **Yes** to delete and **No** to return to the tab.

Category Items

Category Items are used to get or set a specific instance of a variable. There are six Get/Set Category Items:

- Get Category Item (Use Current Path)
- Set Category Item (Use Current Path)
- Get Category Item (All Available)
- Set Category Item (All Available)
- Get Ranked Category Item
- Set Ranked Category Item

All Category items can utilize the Where Clause, conditions and parentheses.

Get Category Item Step

A get category item step is used to get a specific instance of a variable. For example, a get category item step could be used to get the first endorsement for each vehicle on an auto policy.

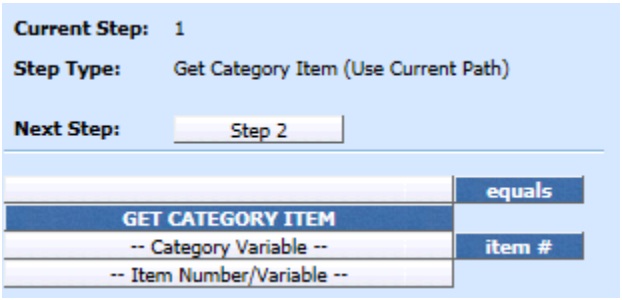


Figure 292 Get Category Step

There are two types of the get category item step.

- **Get Category Item (Use Current Path):** Items in a subcategory of the current category are looked at independently, for each instance of the current category.

  Example - Suppose for an auto policy, you have three drivers, each with some number of violations, as illustrated below:

  <Driver>
  Driver #1
  <Violation>
  Violation Type: Speeding
  Violation Date: 01/04/2011
  Violation Points: 2
  </Violation>
  <Violation>
  Violation Type: Driving without a license
  Violation Date: 05/30/2012
  Violation Points: 1
  </Violation>
  </Driver>

<Driver>
Driver #2
<Violation>
Violation Type: At fault accident
Violation Date: 09/19/2011
Violation Points: 5
</Violation>
</Driver>

<Driver>
Driver #3
<Violation>
Violation Type: Failure to signal
Violation Date: 12/02/2014
Violation Points: 1
</Violation>
<Violation>
Violation Type: Speeding
Violation Date: 01/21/2015
Violation Points: 2
</Violation>
<Violation>
Violation Type: Running a stop sign
Violation Date: 11/14/2015
Violation Points: 2
</Violation>
</Driver>

If a **Get Category Item (Use Current Path)** step, used at the Driver level, is used to get the 2nd violation for each driver.

Driver Second Violation Type [=] Violation Type [Item #] 2

The Driver Second Violation Type would equal "Driving without a license" for Driver #1. For Driver #2, Driver Second Violation Type would be blank, because Driver #2 does not have a second violation. For Driver #3, Driver Second Violation Type would equal "Speeding".

- **Get Category Item (All Available):** Items in a subcategory of the current category are looked at as a whole.

Example - Suppose for an auto policy, you have three drivers, each with some number of violations, as illustrated below:

<Driver>
Driver #1
<Violation>
Violation Type: Speeding
Violation Date: 01/04/2011
Violation Points: 2
</Violation>
<Violation>
Violation Type: Driving without a license
Violation Date: 05/30/2012
Violation Points: 1
</Violation>
</Driver>

<Driver>
Driver #2
<Violation>
Violation Type: At fault accident
Violation Date: 09/19/2011
Violation Points: 5
</Violation>
</Driver>

<Driver>
Driver #3
<Violation>
Violation Type: Failure to signal
Violation Date: 12/02/2014
Violation Points: 1
</Violation>
<Violation>
Violation Type: Speeding
Violation Date: 01/21/2015
Violation Points: 2
</Violation>
<Violation>
Violation Type: Running a stop sign
Violation Date: 11/14/2015
Violation Points: 2
</Violation>
</Driver>

If a **Get Category Item (All Available)** step, used at the Policy level, is used to get the 4th violation on the policy.

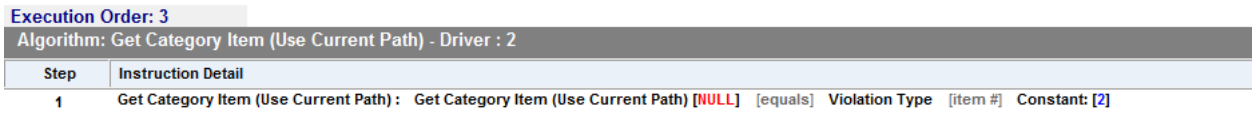
Policy Fourth Violation Type [=] Violation Type [Item #] 4

The Policy Fourth Violation Type would equal "Failure to signal", because that is the fourth violation on the policy.

When Get/Set Conditions are Not Met

When a condition is not met, the system displays a Null value in the debug report for the Get/Set Category item.

For example, if we only wanted the 2nd violation against a driver and “Driver2” has only one Violation type in the current path. The condition we wanted is not met and a Null Value is displayed in the debug report.



This is cast as a zero in calculations.

Set Category Item Step

A set category item step is used to set a specific instance of a variable. For example, a set category item step could be used to set the first accident type for each driver on an auto policy.

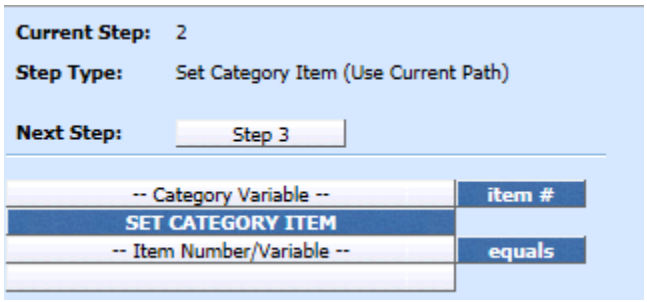


Figure 293 Set Category Item Step

RateManager provides two types of the set category item step.

- **Set Category Item (Use Current Path):** Items in a subcategory of the current category are looked at independently, for each instance of the current category.

  Example

  Suppose for an auto policy, you have three drivers, each with some number of violations, as illustrated below:

  <Driver>
  Driver #1
  <Violation>
  Violation Type: Speeding
  Violation Date: 01/04/2011
  Violation Points: 2
  </Violation>
  <Violation>
  Violation Type: Driving without a license
  Violation Date: 05/30/2012
  Violation Points: 1
  </Violation>
  </Driver>

  <Driver>
  Driver #2
  <Violation>
  Violation Type: At fault accident
  Violation Date: 09/19/2011
  Violation Points: 5
  </Violation>
  </Driver>

  <Driver>
  Driver #3
  <Violation>
  Violation Type: Failure to signal
  Violation Date: 12/02/2014

Violation Points: 1
</Violation>
<Violation>
Violation Type: Speeding
Violation Date: 01/21/2015
Violation Points: 2
</Violation>
<Violation>
Violation Type: Running a stop sign
Violation Date: 11/14/2015
Violation Points: 2
</Violation>
</Driver>

If a **Set Category Item (Use Current Path)** step, used at the Driver level, is used to set the points for the 1st violation for each driver.

Violation Points [Item #] 1 [=] 5
The Violation Points would equal 5 for each driver's 1st violation.

- **Set Category Item (All Available):** Items in a subcategory of the current category are looked at as a whole.

Example

Suppose for an auto policy, you have three drivers, each with some number of violations, as illustrated below:

<Driver>
Driver #1
<Violation>
Violation Type: Speeding
Violation Date: 01/04/2011
Violation Points: 2
</Violation>
<Violation>
Violation Type: Driving without a license
Violation Date: 05/30/2012
Violation Points: 1
</Violation>
</Driver>

<Driver>
Driver #2
<Violation>
Violation Type: At fault accident
Violation Date: 09/19/2011
Violation Points: 5
</Violation>
</Driver>

<Driver>
Driver #3
<Violation>
Violation Type: Failure to signal
Violation Date: 12/02/2014

```
            Violation Points: 1
        </Violation>
        <Violation>
            Violation Type: Speeding
            Violation Date: 01/21/2015
            Violation Points: 2
        </Violation>
        <Violation>
            Violation Type: Running a stop sign
            Violation Date: 11/14/2015
            Violation Points: 2
        </Violation>
</Driver>
```

If a **Set Category Item (All Available)** step, used at the Policy level, is used to set the points for the 4th violation on the policy.

Violation Points [Item #] 4 [=] 5

The Violation Points would equal 5 for Driver #3's 1st violation, because it is the 4th violation on the policy.

Get Ranked Category Item Step

A get ranked category item step is used to get a specific instance of a variable after the variable has been ranked. For example, if driver accidents have been ranked by accident date, high to low, then a get ranked category item step could be used to get the accident date of the most recent accident for each driver.

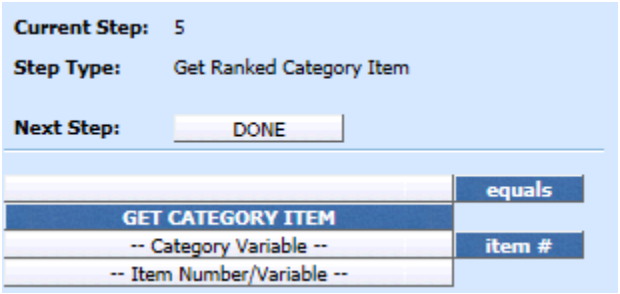


Figure 294 Get Ranked Category

Set Ranked Category Item Step

A set ranked category item step is used to set a specific instance of a variable after the variable has been ranked. For example, if driver accidents have been ranked by claim points, high to low, then a set ranked category item step could be used to set the points for each driver's highest ranked accident to a certain number.

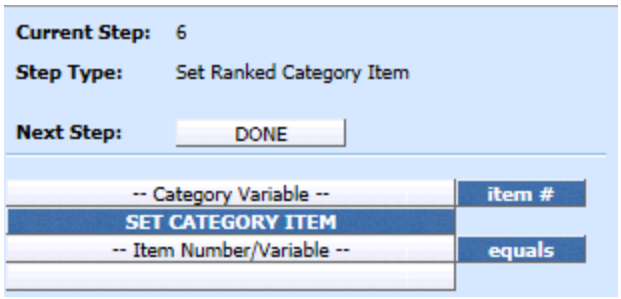


Figure 295 Setting Ranked Category Item

Count across Category Step

A count across category step is used to determine how many instances of a specific item exist. For example, if you wanted to know how many dwellings were on a policy, you could use a count across step to count the total number of instances of a specific dwelling element (field, variable, result).

There are two available in RateManager:

- Count Across Category (All Available)
- Count Across Category (Use Current Path)

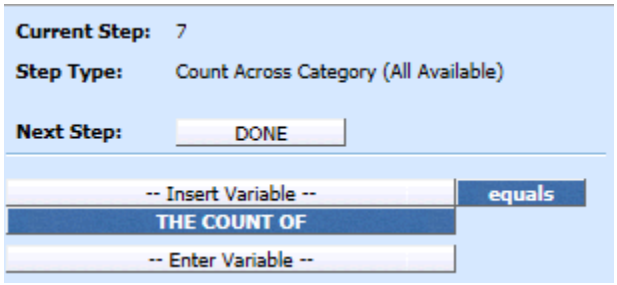


Figure 296 Count Across Category Step

- **Count across Category (Use Current Path):** Items in a subcategory of the current category are looked at independently, for each instance of the current category.

  Example: Suppose for an auto policy, you have three drivers, each with some number of violations, as illustrated below:

  <Driver>
  Driver #1
  <Violation>
  Violation Type: Speeding
  Violation Date: 01/04/2011
  Violation Points: 2
  </Violation>
  <Violation>
  Violation Type: Driving without a license
  Violation Date: 05/30/2012
  Violation Points: 1
  </Violation>
  </Driver>

<Driver>
Driver #2
<Violation>
Violation Type: At fault accident
Violation Date: 09/19/2011
Violation Points: 5
</Violation>
</Driver>

<Driver>
Driver #3
<Violation>
Violation Type: Failure to signal
Violation Date: 12/02/2014
Violation Points: 1
</Violation>
<Violation>
Violation Type: Speeding
Violation Date: 01/21/2015
Violation Points: 2
</Violation>
<Violation>
Violation Type: Running a stop sign
Violation Date: 11/14/2015
Violation Points: 2
</Violation>
</Driver>

If a **Count Across Category (Use Current Path)** step, used at the Driver level, is used to determine the number of violations for each driver.

Total Driver Violations [=] The Count (Using Current Path) of Violation Type

The Total Driver Violations would equal 2 for Driver #1, 1 for Driver #2 and 3 for Driver #3.

- **Count Across Category (All Available):** Items in a subcategory of the current category are looked at as a whole.

Example

Suppose for an auto policy, you have three drivers, each with some number of violations, as illustrated below:

<Driver>
Driver #1
<Violation>
Violation Type: Speeding
Violation Date: 01/04/2011
Violation Points: 2
</Violation>
<Violation>
Violation Type: Driving without a license
Violation Date: 05/30/2012
Violation Points: 1

</Violation>
</Driver>

<Driver>
Driver #2
<Violation>
Violation Type: At fault accident
Violation Date: 09/19/2011
Violation Points: 5
</Violation>
</Driver>

<Driver>
Driver #3
<Violation>
Violation Type: Failure to signal
Violation Date: 12/02/2014
Violation Points: 1
</Violation>
<Violation>
Violation Type: Speeding
Violation Date: 01/21/2015
Violation Points: 2
</Violation>
<Violation>
Violation Type: Running a stop sign
Violation Date: 11/14/2015
Violation Points: 2
</Violation>
</Driver>

If a **Count Across Category (All Available)** step, used at the Policy level, is used to determine the total number of violations on the policy.

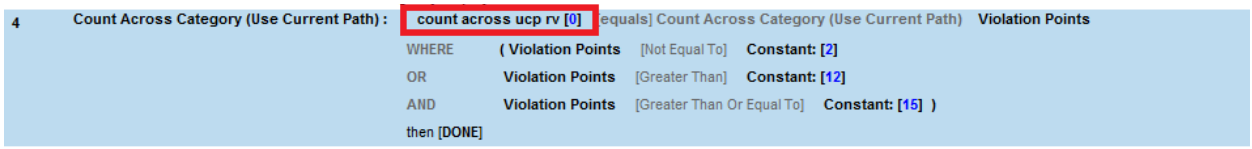
Total Policy Violations [=] The Count (Across All) of Violation Type

The Total Policy Violations would equal 6, because there are six violations on the policy.

When Count Across Conditions are Not Met

When a condition is not met, the system displays a zero (0) value in the debug report.

For example, if we only wanted the 2nd violation against a driver and “Driver2” has only one Violation type in the current path. The condition we wanted is not met and a zero (0) is displayed in the debug report.



The zero value is used in calculations.

Data Type Functions

Data Type functions are an “If step” that checks if a value is numeric, alphabetic, or a date. There are three available in RateManager:

- IsAlpha
- IsDate
- IsNumeric

IsAlpha Step

The IsAlpha step checks if a variable holds an alpha character value. Operators for this option are equals or not equals.

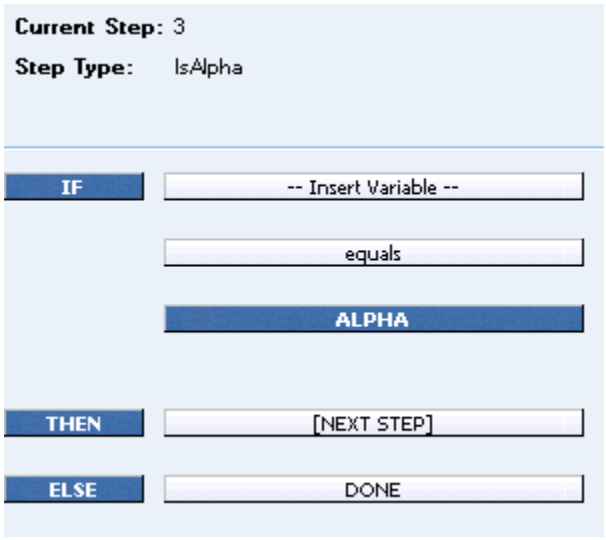


Figure 297 IsAlpha Step

IsDate Step

The IsDate step checks if a variable holds a date value. The date value will be checked using the date format of the program. Operators for this option are equals or not equals.

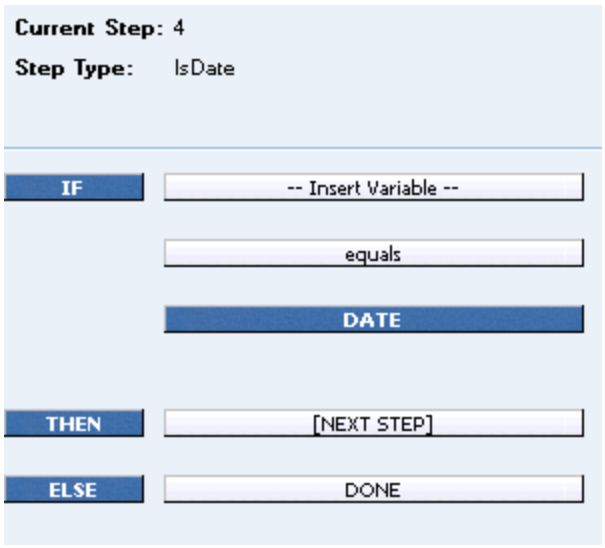


Figure 298 IsDate Step

IsNumeric Step

The IsNumeric step checks if a variable holds a numeric value. The numeric value will be checked using the numeric formats of the program. Operators for this option are equals or not equals.

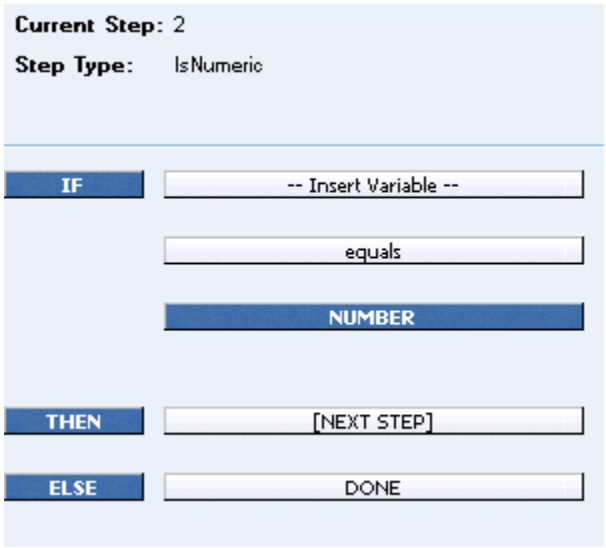


Figure 299 IsNumeric Step

Date Functions

A Date Function step is a special date calculation that calculates the difference between two dates in days, months or years, or adds to a date. The variables used must be a date data type. There are two types of date functions in RateManager:

- Date Addition

- Date Difference
  - Days
  - Months
  - Years

Date Addition Step

A date addition step is used to add a specific number of days, months or years to an existing date. The variables used must be a date data type.

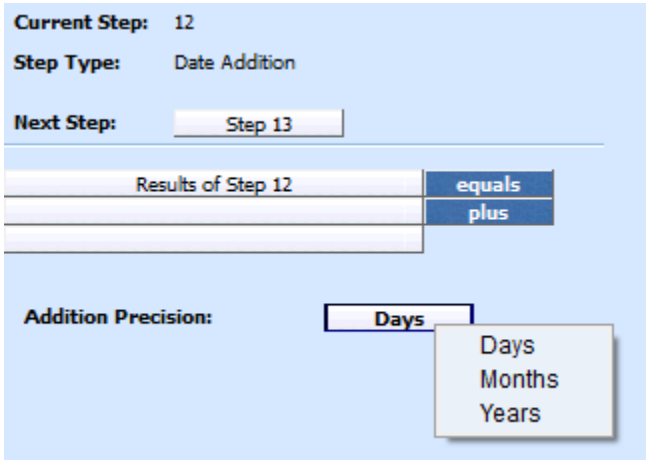


Figure 300 Date Addition

To change the precision used in the addition, click the **Days** box next to **Addition Precision** and select the appropriate precision from the menu.

Date Difference Step

A date difference step is used to determine the difference between two dates. The variables used must be a date data type. For example, if you wanted to determine a driver's age, you would use a date difference step and subtract the driver's birth date from the policy effective date.

Figure 301 Date Difference

RateManager provides three types of the date difference step.

- **Date Difference (days):** The difference is calculated in terms of how many days have elapsed between the two dates, including leap days. For example, 8/12/2003 - 10/15/1999 would result in 1397, because 1397 days have elapsed between these two dates, including the leap day on 2/29/2000.
- **Date Difference (months):** The difference is calculated in terms of how many whole months have elapsed between the two dates. For example, 8/12/2003 - 10/15/1999 would result in 46, because only 46 whole months have elapsed between these two dates.
- **Date Difference (years):** The difference is calculated in terms of how many whole years have elapsed between the two dates. For example, 8/12/2003 - 10/15/1999 would result in 3, because only 3 whole years have elapsed between these two dates.

Signed and Unasigned Results

The Date Difference step returns a signed or unsigned result. A signed result returns a negative value and displays a minus sign for values that are negative.The signed result can be used as criteria of a rating table without having to add additional steps to sign it by comparing the dates. In the debug report, only signed results that are negative will display a designation. Positive values do not have a designation.

Use signed when you want or need to use a negative result. An unsigned result is an absolute number without any designation. The number returned could be positive or negative but the designation is stripped out and is assumed to be positive. Unsigned is the default selection.

To return a signed value, click the Difference Sign option and select Signed.

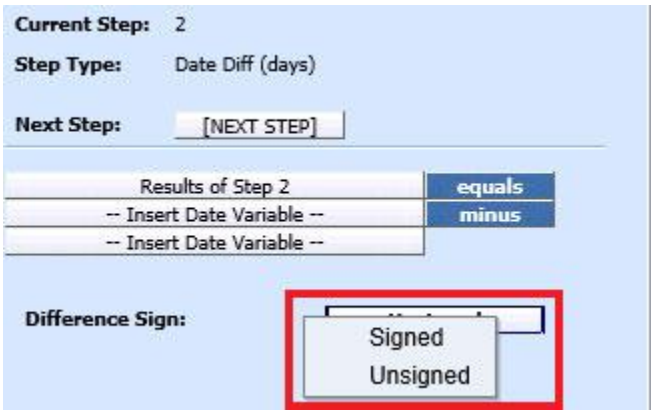


Figure 302 Difference Sign Selection

The Difference Sign selection can be updated at any time. For packaged programs, changes are reflected after the program has been re-packaged.

IF STATEMENTS

IF Statements are used to test criteria and execute certain steps based on whether the statement evaluates to true or false. There are seven types of IF Statements available in RateManager:

- **IF (Normal):** Evaluates to TRUE if the statement is true for the current instance of the element and FALSE otherwise. Can be used for an element in the current category and the current category's parent categories. Can be a simple or complex IF statement.

- **IF All (All Available):** Evaluates to TRUE if the statement is true for all instances of the element and FALSE otherwise. Can be used for an element in any category. Can be a simple or multiple AND condition IF statement. This statement type must use AND conditions only.
- **IF No (All Available):** Evaluates to TRUE if the statement is not true for all instances of the element and FALSE otherwise. Can be used for an element in any category. Can be a simple or multiple OR condition IF statement. This statement type must use OR conditions only.
- **IF Any (All Available):** Evaluates to TRUE if the statement is true for any instance of the element and FALSE otherwise. Can be used for an element in any category. Can be a simple or multiple OR condition IF statement. This statement type must use OR conditions only.
- **IF All (Use Current Path):** Evaluates to TRUE if the statement is true for all instances of the element and FALSE otherwise. Can be used for an element in the current category's child categories. Can be a simple or multiple AND condition IF statement. This statement type must use AND conditions only.
- **IF No (Use Current Path):** Evaluates to TRUE if the statement is not true for all instances of the element and FALSE otherwise. Can be used for an element in the current category's child categories. Can be a simple or multiple OR condition IF statement. This statement type must use OR conditions only.
- **IF Any (Use Current Path):** Evaluates to TRUE if the statement is true for any instance of the element and FALSE otherwise. Can be used for an element in the current category's child categories. Can be a simple or multiple OR condition IF statement. This statement type must use OR conditions only.

IF statements are designed as:

- **Simple IF Statement:** An IF statement with no conditions. Any IF type can be used as a simple IF statement. Any IF statement can be a simple IF statement.
- **Multiple IF Statement:** A string of IF statements that uses the same Condition type throughout the entire statement.
  - IF All (All) and IF All (UCP) are AND statements only. You must use all AND conditions.
  - IF No (All), IF No (UCP), IF Any (All), and IF Any (UCP) are OR statements only. You must use all OR conditions.
- **Complex IF Statement:** A string of IF statements that uses both Condition types in the statement. Only the IF (Normal) can be a complex IF statement.

Then/Else Terms in IF Statements

Each IF Statement contains a THEN term and an ELSE term. When running the step, any IF Statement that evaluates to TRUE will run the THEN term and any IF Statement that evaluates to FALSE will run the ELSE term.

If conditions are used, when the parameters are met, the statement is evaluated to be true. If the parameters are not met, the statement is evaluated to be false. This does not mean that the data is evaluated to be true. Only that the parameters used to define the data have been met.

Complex IF

An IF (Normal) statement that contains both OR conditions and AND conditions is a complex IF statement. When the step is run, any AND condition must be met along with at least one OR condition in order for a true value to be returned. When a true value is returned, the THEN part of the statement will be run. If a false value is returned by the IF Statement, the ELSE part of the statement will be run.

An IF (Normal) statement can make use of parentheses when both OR and AND conditions are added.

For example: If you wanted to execute a statement where Variable A is greater than zero, OR Variable B is less than 1000, AND Variable C [=] 100 then continue, else stop.

IF Variable A [>] O
OR
IF Variable B [<] 1000
AND
IF Variable C [=] 100

(If the OR condition is **true** and the AND condition is **true**) THEN Step 4

(If the OR condition is **false** and the AND condition is **true**) ELSE Done

(If the OR condition is **false** and the AND condition is **false**) ELSE Done

Only IF (Normal) can be used for complex IF statements.

Entering an IF Statement

An IF Statement can be entered at any time in Calculated Variables, Filter Rules, Driver Assignments, and Algorithms.

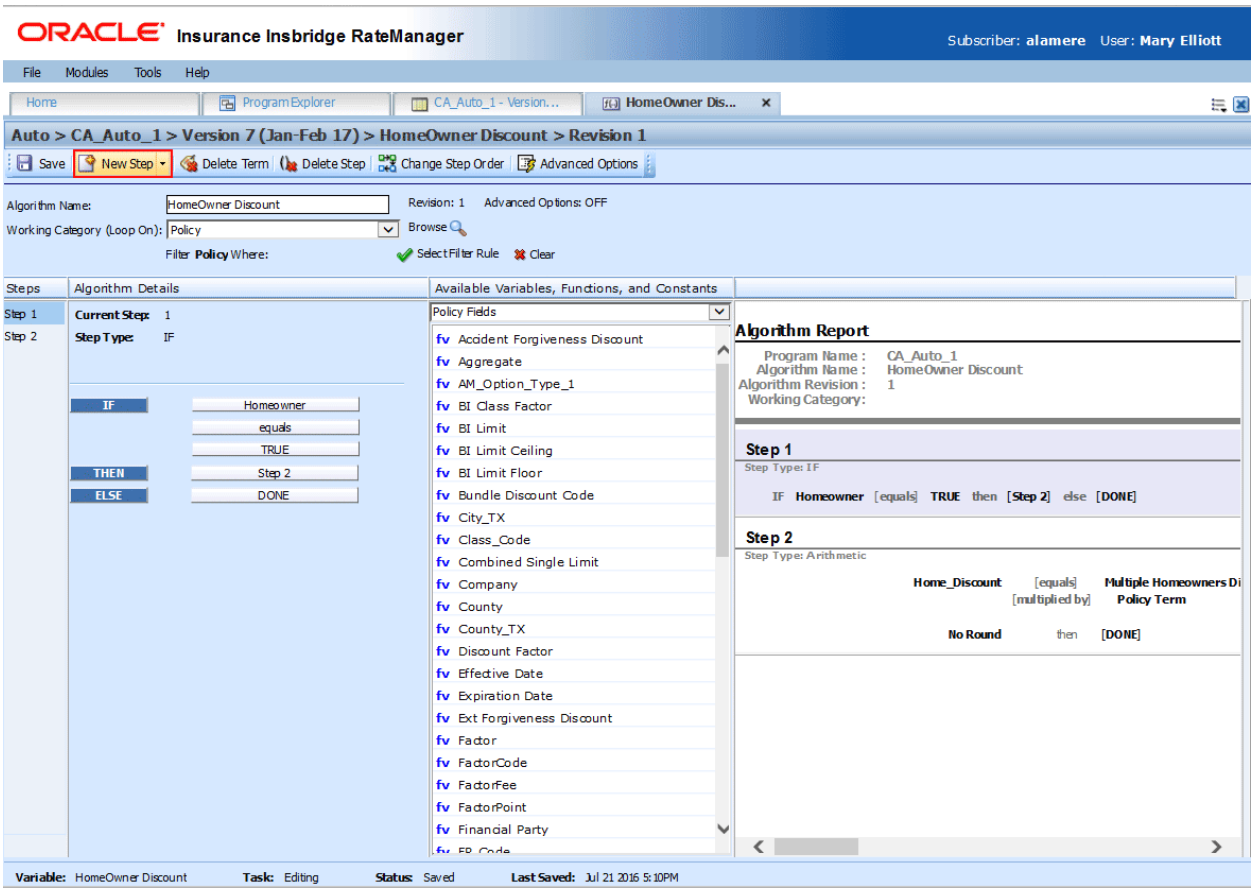


Figure 303 Creating a Simple IF Statement

1. Select your New **IF Step Type** from the New Step option at the top of the screen. The screen refreshes with a simple IF statement.

2. Select the term you want to complete. The field is highlighted.

3. Select the term you want in the field from the Available Variables, Functions and Constants. The field is completed with the term.

4. To add a condition, click the **IF blue box**. An option menu is displayed. Click **Insert** and select to Insert an AND condition or an OR condition. The screen refreshes with your selection directly under the IF statement. Enter in your criteria.

5. To add more conditions, you can select the IF blue box and add a condition directly under the IF statement. This pushes the first condition down the list. You also can click the condition blue box and select to insert an AND condition or an OR condition under the existing condition. This leaves the first condition in the first position and places the newer condition in the second position. The screen refreshes with your selection. Enter in your criteria.

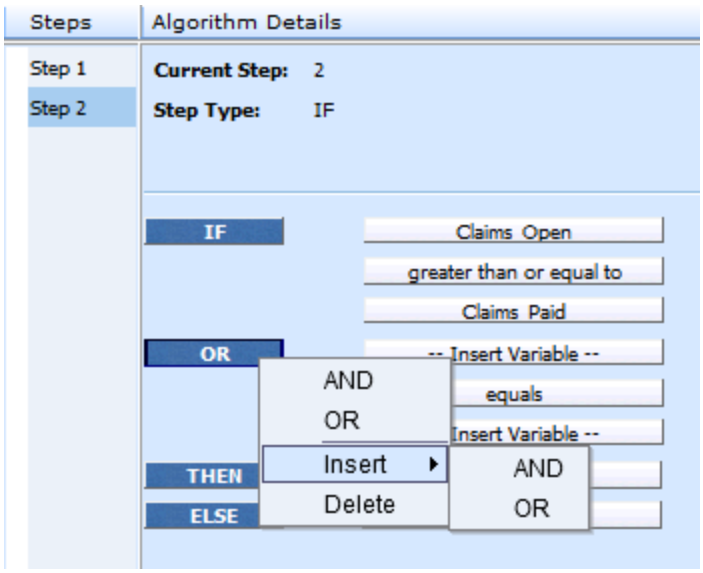


Figure 304 Inserting Conditions in an IF Statement

You can change the condition, insert additional conditions or delete a condition statement by clicking the condition (AND or OR) blue box and selecting from the menu.

Options are:

- **AND:** The AND in the upper portion of the menu changes the selected condition to an AND type condition.
- **OR:** The OR in the upper portion of the menu changes the selected condition to an OR type condition.
- **Insert AND:** Inserts an AND condition under the selected condition.
- **Insert OR:** Inserts an OR condition under the selected condition.
- **Delete:** Deletes the selected condition. No warning is given. The action is immediate.

NOTE: *For details on the options available to each If Statement type, please see IF Statements.*

6. Continue until you have all your conditions entered. Complete the THEN and ELSE statements. And save your work

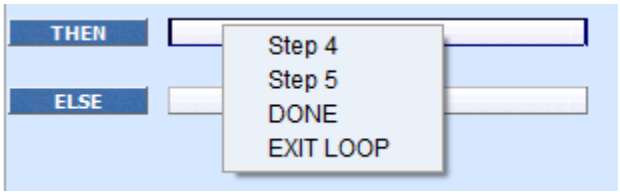


Figure 305 Then and Else Segments of an IF Statement

Updating an IF Statement

The IF statement step type at the top of the step applies to the entire IF statement. You can update an IF statement and change to a different type of IF Statement by clicking the blue IF box and selecting a different IF Statement from the menu.

The screen refreshes and the updated step type is listed at the top of the step.

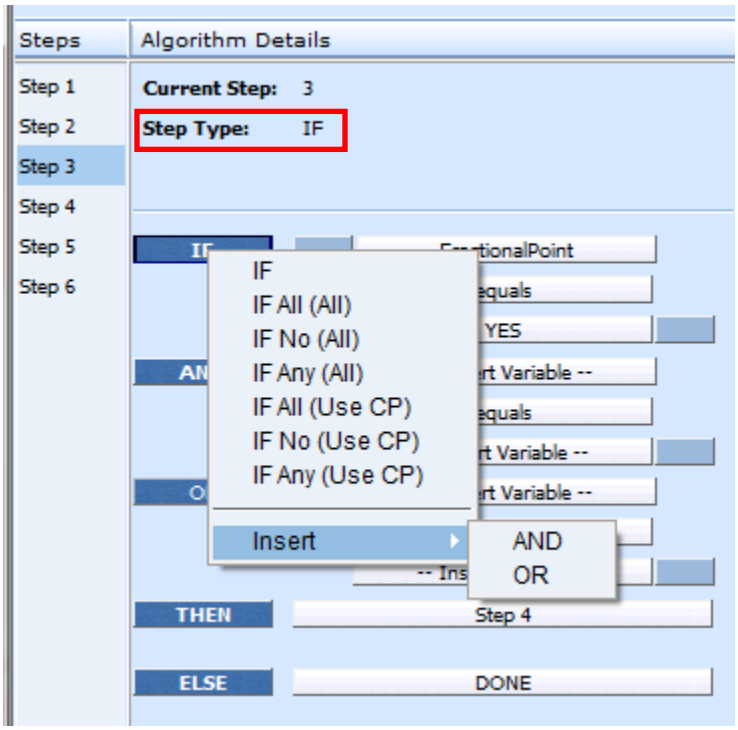


Figure 306 Editing a Multiple If Statement

You also can add conditions from the same menu or from the menu on existing terms. Conditions will be populated directly under the statement. The Then and Else segment will be pushed down the list.

Deleting a Condition from an IF Statement

You can delete a condition you do not need. Right click the AND or the OR blue box of the condition you want to delete. A menu will be displayed. Click **Delete**. The screen refreshes and the condition are removed.

Mask Step

A mask step is used when you want to use a part of a string. For example, if a vehicle's ID number is passed into the system as VEH01, but you only wanted the last two digits, you could use a mask step to store those digits in another element.

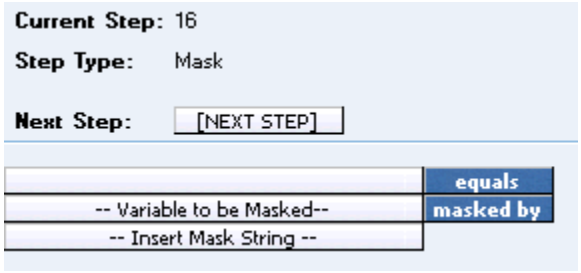


Figure 307 Mask Step

Available Mask Options

Function	Definition	Keystroke(s)
Hold	Holds a character, i.e. keeps the character	~
Remove	Replaces a character with another character	|
Replace	Removes a character Insert adds a character	any character
Insert	Adds a character.	^ + any character

Figure 308 Available Mask Options Table

Examples

Function	Input	Mask	Result
Hold	GA263SX4597	~~~~~	GA263
Remove	VEH01	|||~~	01
Replace	Insbridge	~~~urance	Insurance
Insert	Cat	~^h~^r~	Chart

Figure 309 Mask Option Examples Table

Math Functions

Math Functions are used when you want to perform mathematical operations. Most often this involves creating an equation where a single numeric value can be obtained from plugging in other numeric values and arithmetic operators. Math Functions also allow you to remove any negative values so that you can work with positive values. Math Functions has thirteen options:

- Arithmetic Step

- Absolute Value
- Power
- Log
- Log10
- Natural Log
- Exp
- Rand
- Fact
- Sqrt
- Ceiling
- Floor
- Even
- Odd

Arithmetic Step

An arithmetic step is used to perform basic arithmetic, addition, subtraction, multiplication and division. It also can be used to perform bitwise OR and bitwise AND operations. For more information, see Adding a New Element.

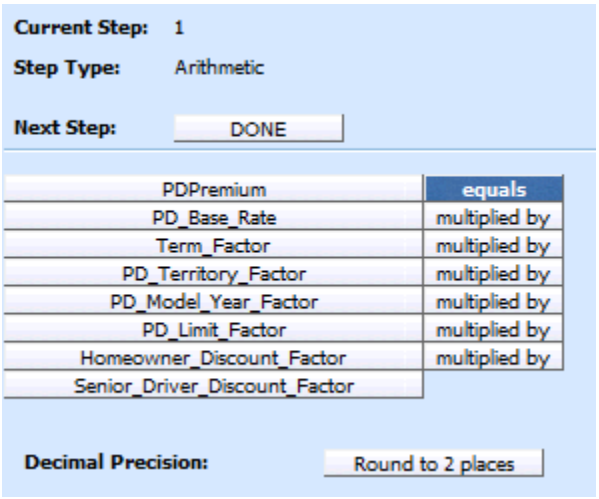


Figure 310 Arithmetic Step

Adding a New Element

To add a new element, a variable, field, result or constant, to the step, find the item you would like to add from the listing of **Available Variables, Functions and Constants.** Once you have located the correct item, double-click it to add it as the next operand.

The Operator must be selected before the step can be saved.

You also can replace an existing operand by clicking it once and then double clicking the item you would like to replace it with. This applies to the results of the arithmetic step too.

Adding Arithmetic Operators

To set the arithmetic operator to be used, click the box next to the item name and select the appropriate operator from the menu.

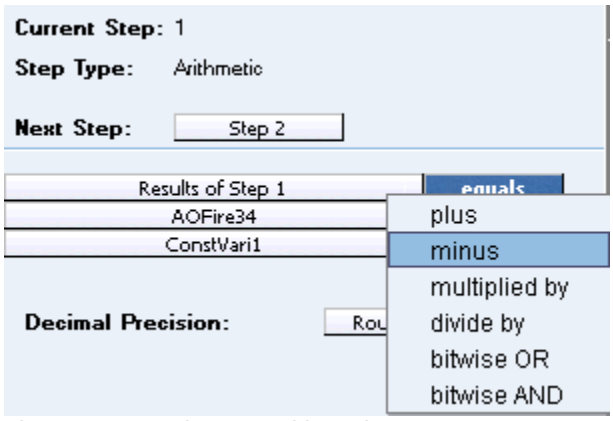


Figure 311 Setting an Arithmetic Step 2

Arithmetic Operators

Arithmetic operators are used in arithmetic steps. The available arithmetic operators are:

+	Plus - Performs a basic addition operation. i.e. 2 + 3 = 5.
-	Minus - Performs a basic subtraction operation, i.e. 5 - 3 = 2.
x	Multiply By - Performs a basic multiplication operation, i.e. 2 x 3 = 6.
/	Divide By - Performs a basic division operation, i.e. 6 / 3 = 2.
|	Bitwise OR - Performs a bitwise OR operation, i.e. 5 | 12 = 13.
&	Bitwise AND - Performs a bitwise AND operation, i.e. 5 & 12 = 4.

Bitwise OR

Bitwise OR is determined by converting each number to binary form and then performing an OR, bit by bit, where the value returns as a 0 or 1. A 0 value is returned in the result if a 1 is not in any row of the column being evaluated. A 1 is returned if a 1 is in any row of the column being evaluated. For example:

5 in binary is:	00000101
12 in binary is:	00001100
So the result is:	00001101 or 13.

Any column that has a 1 in either row has a 1 in the result row.

Bitwise AND

Bitwise AND is determined by converting each numbers to binary form and then performing an AND, bit by bit, where the value returns as a 0 or 1. A 0 value is returned in the result if a 1 is not in every row of the column being evaluated. A 1 is returned if a 1 is in every row of the column being evaluated. For example:

5 in binary is:	00000101
12 in binary is:	00001100
So the result is:	00000100 or 4.

Any column that has a 1 in each row will have a 1 in the result row.

Absolute Value Step

Get Absolute Value Of step gets the absolute value of the selected variable. The absolute value of a number x, commonly written as |x|, is the number x without regard to sign. For example, the absolute value of -8 (negative 8) is 8.

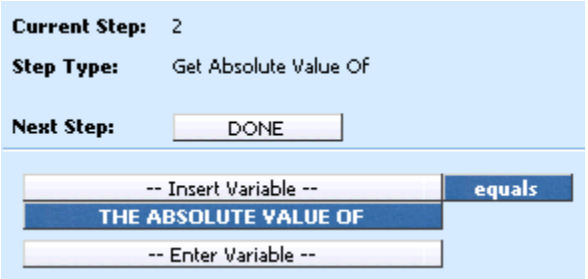


Figure 312 Absolute Value

Power Step

A power step is used when you want to raise a number to a power. The components of the power step are Result, Base, and Power. For the result component, you can choose to use a Result Variable or a Result of Step Variable. For the base component, you can choose to use any variable (with a numeric data type) or a constant value. For the power component, you can choose to use any variable (with a numeric data type) or a constant value.

Example Use Case:

If you wanted to use experience based information (historical claims and/or exposure numbers) as part of your rating logic and you wanted to account for inflation/deflation, then you could use the power function to perform the inflation/deflation calculations. If you had claims information from an insured for $10,000, you could estimate that in 5 years' time the claim value might increase by 3% a year. This could be represented by $10,000 x 1.03^5 = $11,592.

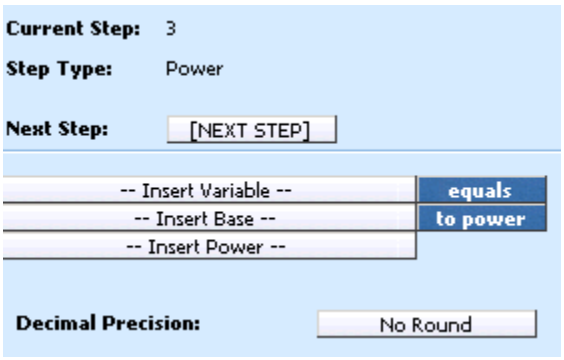


Figure 313 Power Step

Log Step

The log step returns the logarithm of a number to the base you specify. You must enter the number and the base.

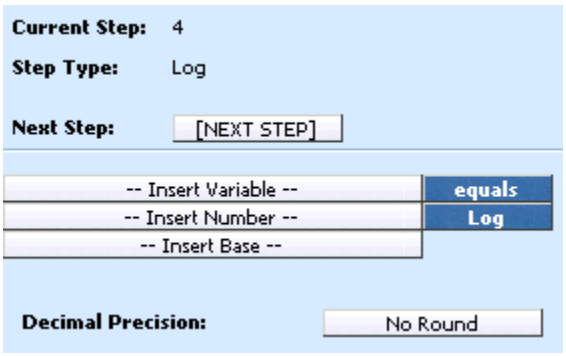


Figure 314 Log Step

Log10 Step

The log10 step returns the base-10 logarithm of a number.

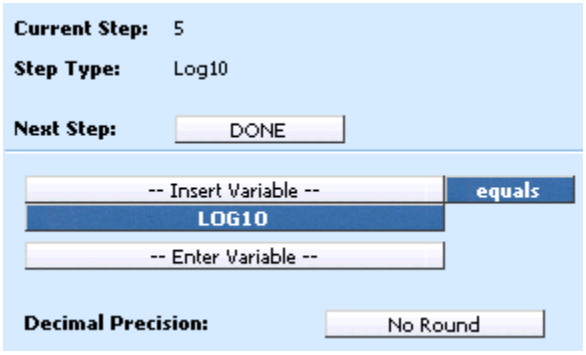


Figure 315 Log10 Step

Natural Log

The natural log step returns the natural logarithm of the value of –Enter Number-- (that is, the logarithm to the base e, where e is the irrational transcendental constant with the approximate value of 2.718281828). The value for Enter Number must be a number data type, for example integer, decimal, or real – also, the value of Enter Number cannot be zero and cannot be negative.

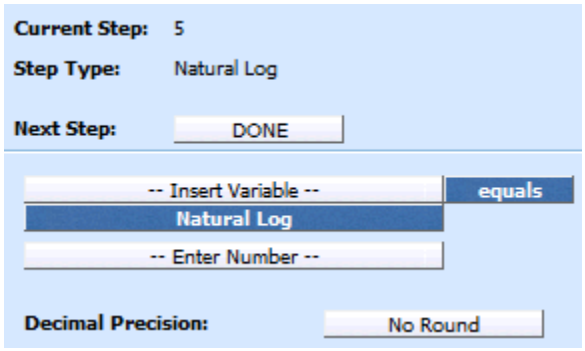


Figure 316 Natural Log Step

Exponent Step

The exponent step allows for a constant *e* to be raised to the power of a number. The constant *e* equals 2.71828182845904, the base of the natural logarithm.

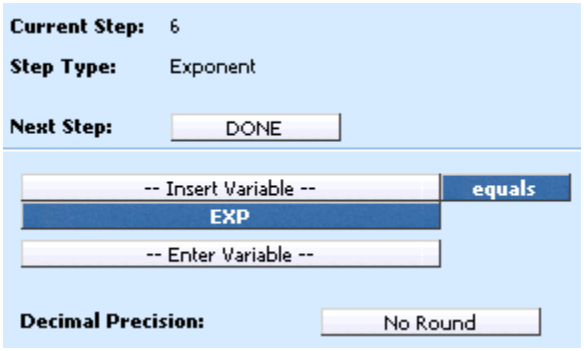


Figure 317 Exponent Step

Rand Step

The random (rand) step returns an evenly distributed random number, greater than or equal to 0, and less than 1. A new random number is returned for every time the random step is used. The variable used must be a number data type, for example integer or decimal.

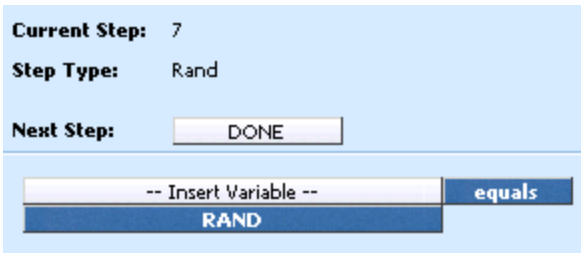


Figure 318 Random Number Step

Fact Step

The factorial (fact) step returns the factorial of a number. Number values must be positive. The variable used must be a number data type, for example integer or decimal.

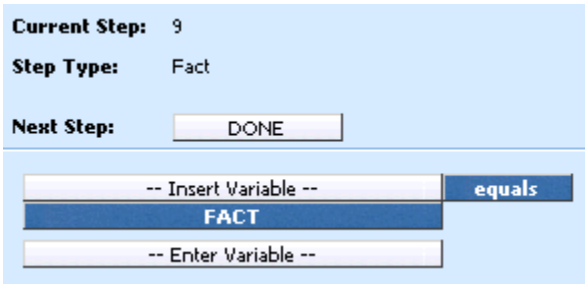


Figure 319 Factorial Step

Square Root Step

The square root step returns a positive square root of the number entered. If the number is negative, an error is shown. The variable used must be a number data type, for example integer or decimal.

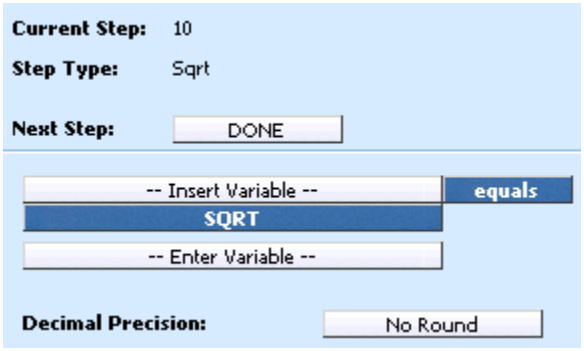


Figure 320 Square Root Step

Ceiling Step

The ceiling step allows for a number to be rounded up away from zero, to the nearest multiple of significance. For example, CEILING(5.68, .5) would result in the ceiling for the number (5.68) being rounded up away from zero by a factor of (.5). Number and significance must be the same sign. You cannot set the number to a negative value and the significance to a positive value. You will need both a number and significance to use this step type.

Example Use Case:

If you want to avoid using pennies in your prices and your product is priced at $4.42, use the formula =CEILING(4.42,0.05) to round prices up to the nearest nickel.

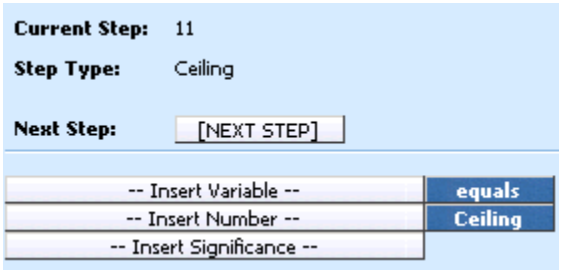


Figure 321 Ceiling Step

Floor Step

The floor step allows for a number to be rounded down towards zero, to the nearest multiple of significance. For example, FLOOR(5.68, .5) would result in the floor for the number (5.68) being rounded down to the nearest zero by a factor of (.5). Number and significance must be the same sign. You cannot set the number to a negative value and the significance to a positive value. You will need both a number and significance to use this step type.

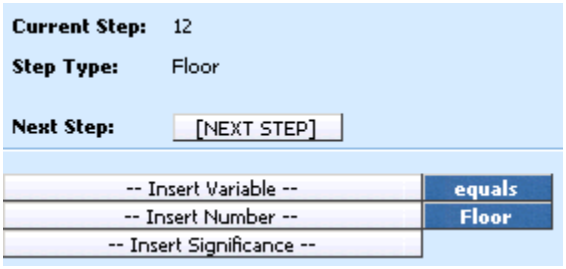


Figure 322 Floor Step

Even Step

The even step allows for a number to be rounded up to the nearest even integer. The variables used must be a number data type, for example integer or decimal.

Figure 323 Even Step

Odd Step

The odd step allows for a number to be rounded up to the nearest odd integer. The variables used must be a number data type, for example integer or decimal.

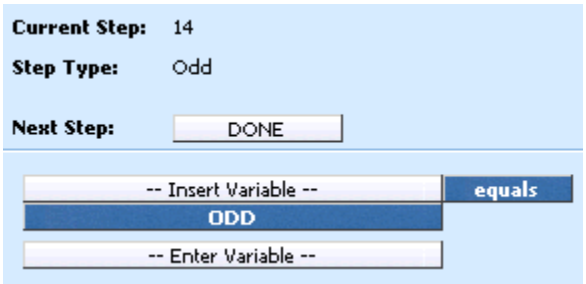


Figure 324 Odd Step

Trig Functions

RateManager has fourteen trigonometry functions available to algorithms, calculated variables and driver assignments.

- Cos
- Cosh
- Acos
- Acosh
- Sin
- Sinh
- Asin
- Asinh
- Tan
- Tanh
- Atan
- Atanh
- Degrees
- Radians

Cos Step

The cos step returns the cosine of the given angle. The variables used must be a number data type, for example integer or decimal.

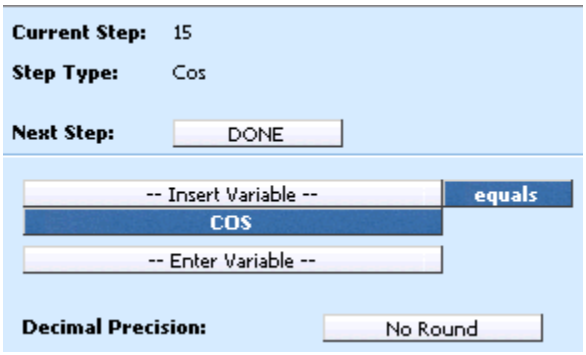


Figure 325 Cosine Step

Cosh Step

The cosh step returns the hyperbolic cosine of a number. The variables used must be a number data type, for example integer or decimal.

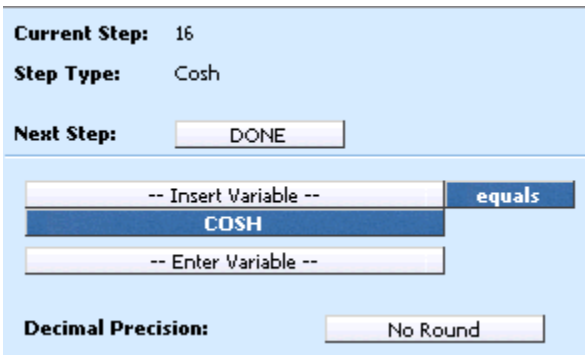


Figure 326 Cosh Step

Acos Step

The acos step returns the arccosine, or inverse cosine, of a number. The variables used must be a number data type, for example integer or decimal.

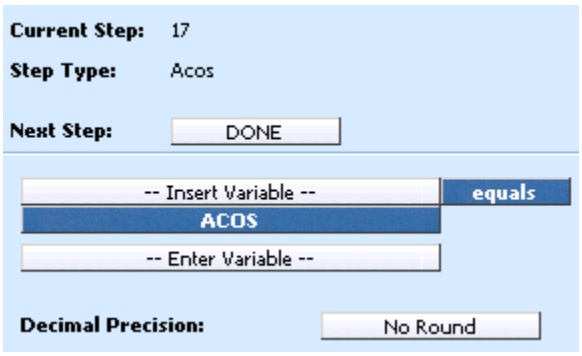


Figure 327 Acos Step

Acosh Step

The acosh step returns the inverse hyperbolic cosine of a number. The variables used must be a number data type, for example integer or decimal.

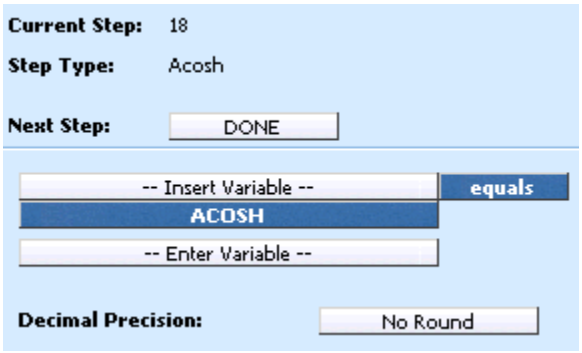


Figure 328 Acosh Step

Sin Step

The sin step returns the sine of the given angle. The variables used must be a number data type, for example integer or decimal.

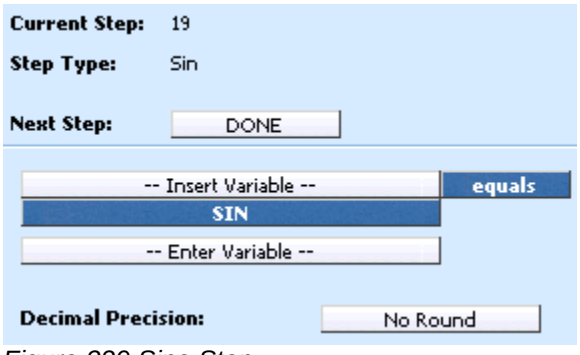


Figure 329 Sine Step

Sinh Step

The sinh step returns the hyperbolic sine of a number. The variables used must be a number data type, for example integer or decimal.

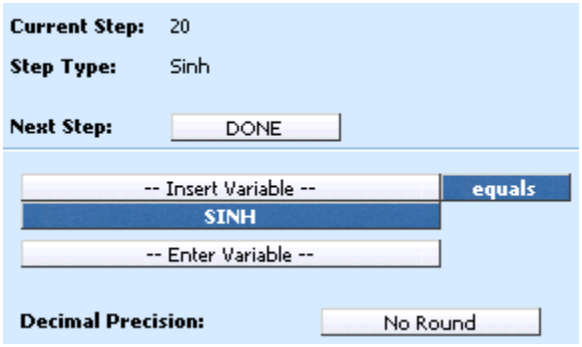


Figure 330 Sinh Step

Asin Step

The asin step returns the arcsine, or inverse sine, of a number. The variables used must be a number data type, for example integer or decimal.

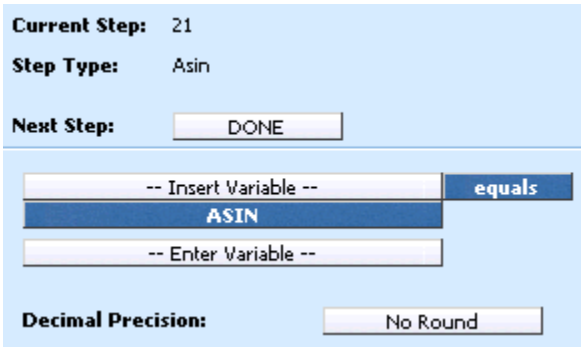


Figure 331 Asin Step

Asinh Step

The asinh step returns the inverse hyperbolic sine of a number. The variables used must be a number data type, for example integer or decimal.

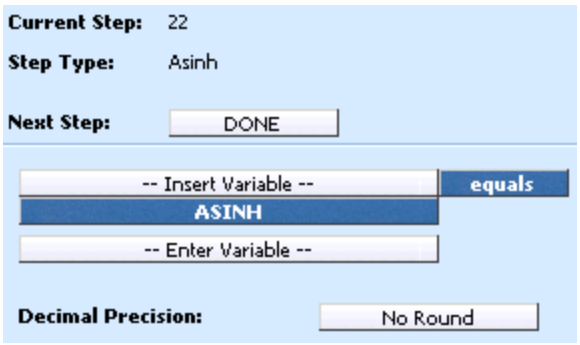


Figure 332 Asinh Step

Tan Step

The tan step returns the tangent of the given angle. The variables used must be a number data type, for example integer or decimal.

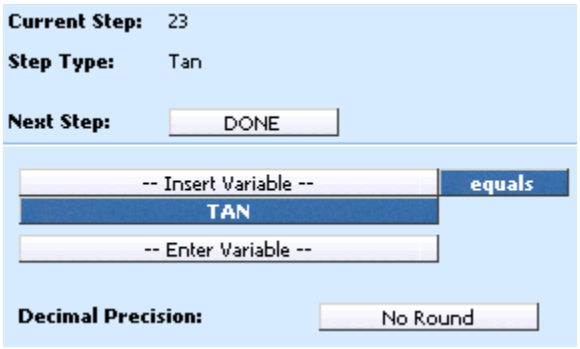


Figure 333 Tangent Step

Tanh Step

The tanh step returns the hyperbolic tangent of a number. The variables used must be a number data type, for example integer or decimal.

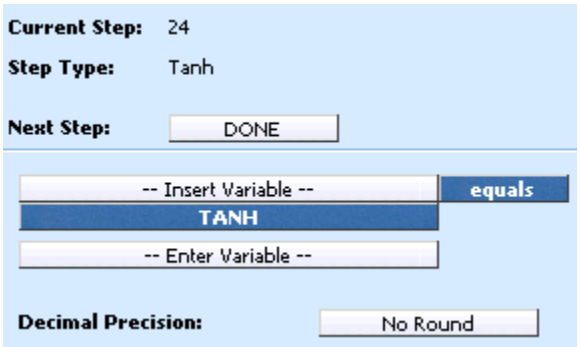


Figure 334 Tanh Step

Atan Step

The atan step returns the arctangent, or inverse tangent, of a number. The variables used must be a number data type, for example integer or decimal.

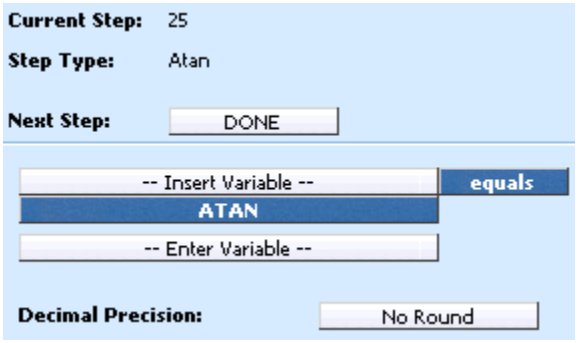


Figure 335 Atan Step

Atanh Step

The atanh step returns the inverse hyperbolic tangent of a number. The variables used must be a number data type, for example integer or decimal.

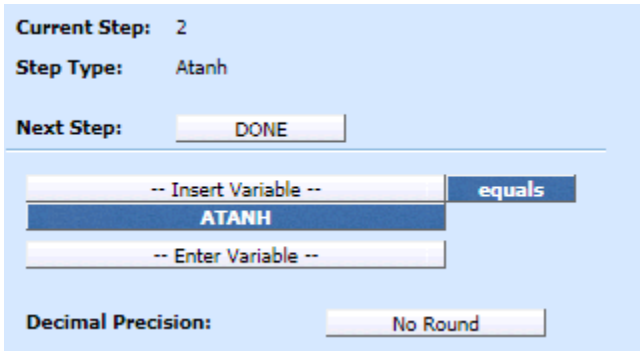


Figure 336 Atanh Step

Degrees Step

The degrees step converts radians into degrees. The variables used must be a number data type, for example integer or decimal.

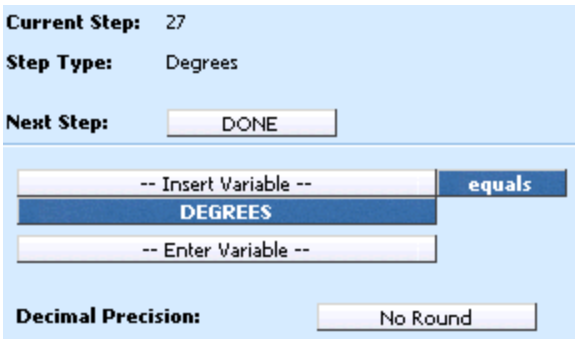


Figure 337 Degrees Step

Radians Step

The radians step convert degrees into radians. The variables used must be a number data type, for example integer or decimal.

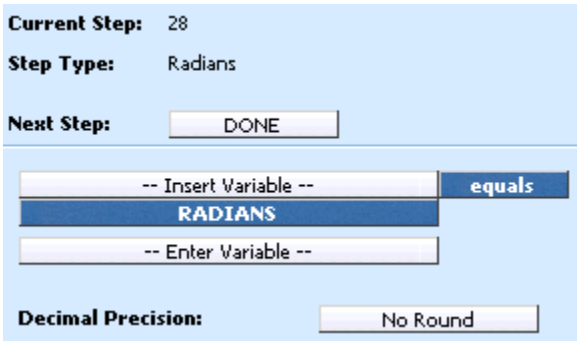


Figure 338 Radians Step

Statistical Functions

RateManager has statistical functions available to algorithms, calculated variables and driver assignments.

- Log Gamma
- Incomplete Gamma
- Gamma

Log Gamma

The log gamma step returns the natural logarithm of the Complete Gamma Function with parameter Alpha. The value for Alpha must be a number data type, for example integer, decimal, or real. Alpha must be greater than zero.

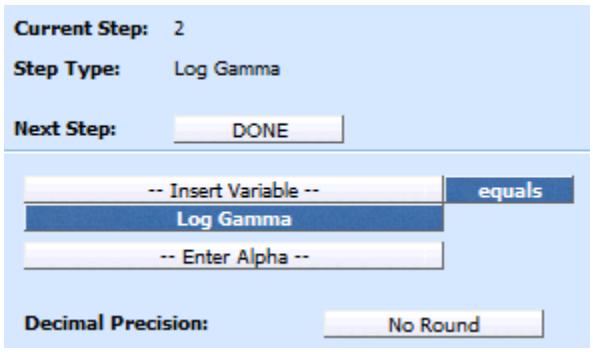


Figure 339 Log Gamma Step

Incomplete Gamma

The incomplete gamma step returns the Lower Incomplete Gamma Function, whose Alpha parameter is the value of --Insert Alpha-- and whose upper bound of integration is the value of --Insert Number--. The value for Insert Alpha must be a number data type, for example integer, decimal, or real. Alpha must be greater than zero. The value of Insert Number must be a number data type also, and it must be greater than or equal to zero.

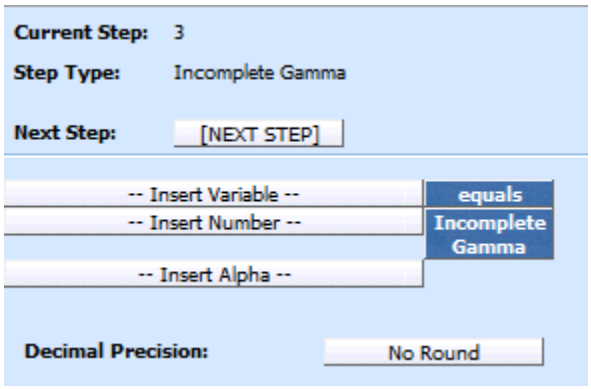


Figure 340 Incomplete Gamma Step

Gamma

The gamma step returns the value of the Complete Gamma Function of the --Enter Alpha-- variable. The value for Alpha must be a number data type, for example integer, decimal, or real. Alpha must be greater than zero.

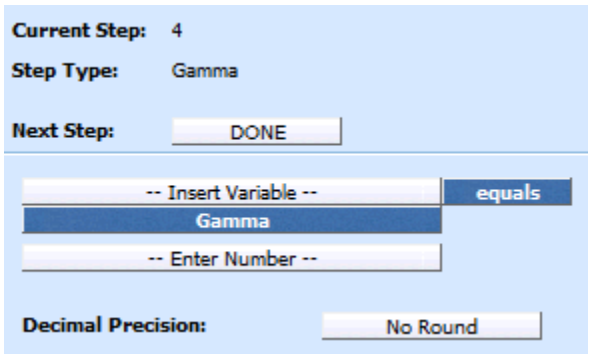


Figure 341 Gamma Step

Ranking

A Rank Across category step ranks all instances of a specific element, either highest to lowest or lowest to highest.

Ranking can use Where clauses, conditions and parentheses.

There are two types of Ranking Steps:

- **Clear Ranking**
- **Rank Across Category**

RateManager provides four types of the Rank across Category Step.

- **Rank across Category (All Available) (Low-to-High):** Ranks elements in any category, lowest to highest.
- **Rank across Category (High-to-Low):** Ranks elements in the current category, the current category's child categories and the current category's parent categories, highest to lowest.
- **Rank across Category (Low-to-High):** Ranks elements in the current category, the current category's child categories and the current category's parent categories, lowest to highest.
- **Rank across Category (All Available) (Low-to-High):** Ranks elements in any category, highest to lowest.

Clear Ranking Step

When ranking a Ranking Step, the system allows for cascading ranking operations. This step breaks the cascading ranking that is inherent when using back-to-back ranking operations. This step should always be used after a ranking step, when cascading is not desired. When in Driver Assignment, it is recommended to a Clear Ranking function to clear a driver assignment ranking.

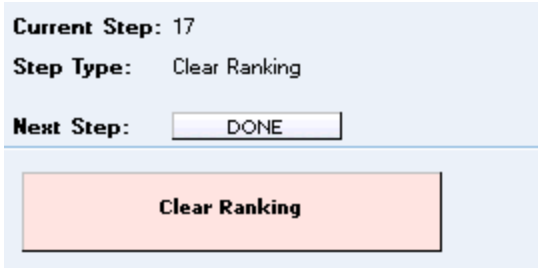


Figure 342 Clear Ranking Step

Rank across Category Step

A rank across category step ranks all instances of a specific element, either highest to lowest or lowest to highest.

You can rank instances of a child category based on a date type variable. The value of variables of instances of a child category ordered in chronological order are available without having to create a calculated variable to calculate the “age” in days to order them.

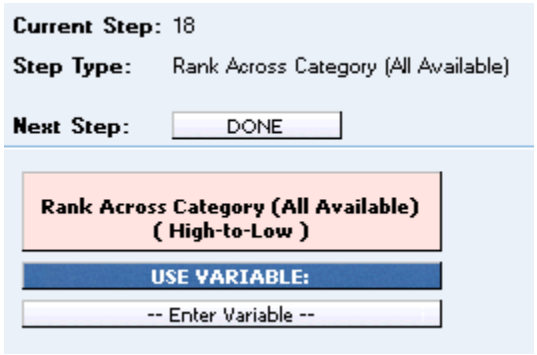


Figure 343 Rank Across Category Step

String Functions

String functions can be used any time you modify a ‘data’ type of string. String Functions include:

- Get Length
- Set Message
- String Addition

Get Length

A Get Length step gets the number of characters in a specified string. You can use the LEN (Get String Length) function for any string variable. In Figure 344, the **Results of the Step** will equal 5, since the constant value of "10000" is 5 characters.

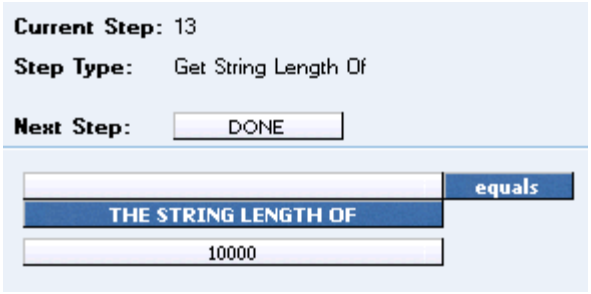


Figure 344 Get Length

Set Message

A Set Message step is used when you want a specific message stored in a string. For example, if you wanted to output the message "Too many claims in last three years." if a policy fails a specific underwriting rule, you could use a Set Message step. The variable field must be a string data type.

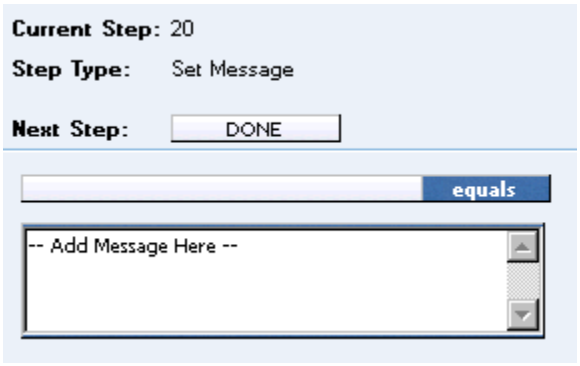


Figure 345 Set Message

The message field can contain up to 1024 characters and allows for these special characters: asterisk (*), colon (:), and question mark (?). See Restrictions for more information.

String Addition

A String Addition step is used to concatenate two or more strings or character together. For example, using addition, the string "Ins" plus the string "bridge" would result in "Insbridge". Multiple field strings are allowed. Select the term you want to add and double click. The only operator will be plus.

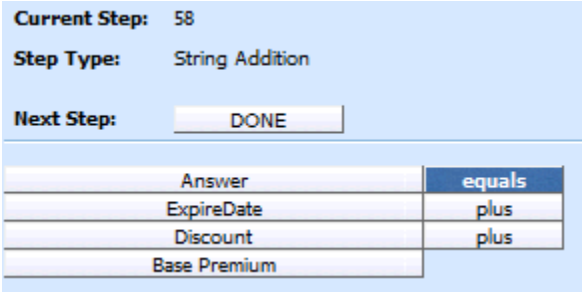


Figure 346 String Addition Step

Sum across Category Step

A sum across category step is used to determine the sum of all instances of a specific element (field, variable, result). For example, if you had a Total Vehicle Premium for each vehicle on an auto policy, and you wanted to know the total policy premium, you would sum across the Total Vehicle Premium.

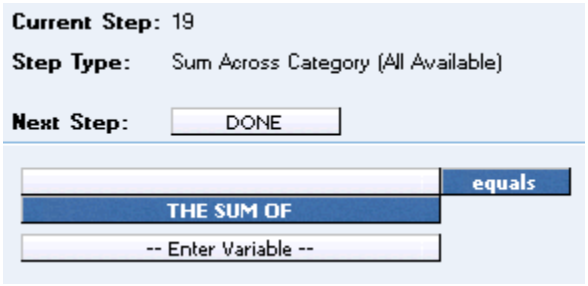


Figure 347 Sum Across Category Step

RateManager provides two types of the Sum Across Category Step.

- **Sum across Category (Use Current Path):** Items in a subcategory of the current category are looked at independently, for each instance of the current category.

  Example

  Suppose for an auto policy, you have three drivers, each with some number of violations, as illustrated below:

  <Driver>
  Driver #1
  <Violation>
  Violation Type: Speeding
  Violation Date: 01/04/2011
  Violation Points: 2
  </Violation>
  <Violation>
  Violation Type: Driving without a license
  Violation Date: 05/30/2012
  Violation Points: 1
  </Violation>
  </Driver>

  <Driver>
  Driver #2
  <Violation>
  Violation Type: At fault accident
  Violation Date: 09/19/2011
  Violation Points: 5
  </Violation>
  </Driver>

  <Driver>
  Driver #3
  <Violation>

Violation Type: Failure to signal
Violation Date: 12/02/2014
Violation Points: 1
</Violation>
<Violation>
Violation Type: Speeding
Violation Date: 01/21/2015
Violation Points: 2
</Violation>
<Violation>
Violation Type: Running a stop sign
Violation Date: 11/14/2015
Violation Points: 2
</Violation>
</Driver>

If a **Sum across Category (Use Current Path)** step, used at the Driver level, is used to determine the number of violation points for each driver.

Total Driver Violation Points [=] The Sum (Using Current Path) of Violation Points

The Total Driver Violation Points would equal 3 for Driver #1, 5 for Driver #2 and 5 for Driver #3.

- **Sum across Category (All Available):** Items in a subcategory of the current category are looked at as a whole.

Example

Suppose for an auto policy, you have three drivers, each with some number of violations, as illustrated below:

<Driver>
Driver #1
<Violation>
Violation Type: Speeding
Violation Date: 01/04/2011
Violation Points: 2
</Violation>
<Violation>
Violation Type: Driving without a license
Violation Date: 05/30/2012
Violation Points: 1
</Violation>
</Driver>

<Driver>
Driver #2
<Violation>
Violation Type: At fault accident
Violation Date: 09/19/2011
Violation Points: 5
</Violation>
</Driver>

<Driver>
Driver #3

<Violation>
Violation Type: Failure to signal
Violation Date: 12/02/2014
Violation Points: 1
</Violation>
<Violation>
Violation Type: Speeding
Violation Date: 01/21/2015
Violation Points: 2
</Violation>
<Violation>
Violation Type: Running a stop sign
Violation Date: 11/14/2015
Violation Points: 2
</Violation>
</Driver>

If a **Sum across Category (All Available)** step, used at the Policy level, is used to determine the total number of violation points on the policy.

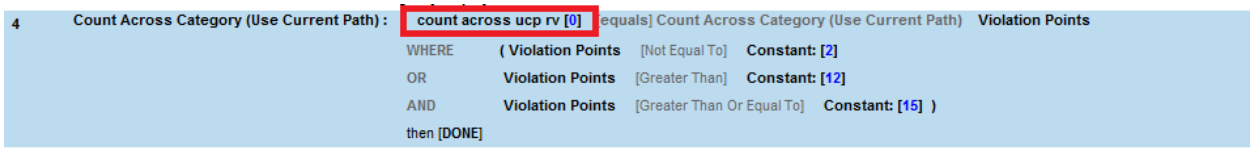
Total Policy Violation Points [=] The Sum (Across All) of Violation Points

The Total Policy Violation Points would equal 13.

When Sum Across Conditions are Not Met

When a condition is not met, the system displays a zero (0) value in the debug report.

For example, if we only wanted the 2nd violation against a driver and "Driver2" has only one Violation type in the current path. The condition we wanted is not met and a zero (0) is displayed in the debug report.



The zero value is used in calculations.

Multiply Across Category

Multiplies values across instances of a child category and functions like the sum across category step. The result can be used in a calculated variable or as a criteria in a table variable. There are two options:

- Use current path
- All available

Where clauses can be added.

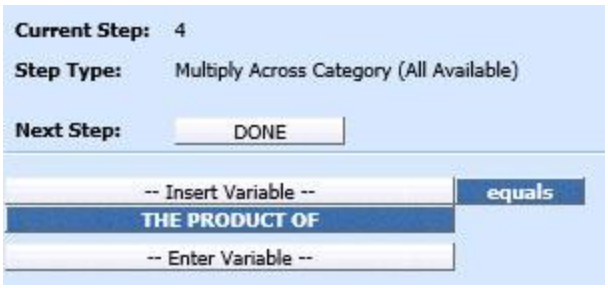


Figure 348 Multiple Across Category Step

Minimum/Maximum Step

A Minimum/Maximum step determines the minimum or maximum value from a list. Integer, decimal and date data types can be used. String values are not allowed. Any type of variable can be used in the list; field, caluculated variable, results, etc. The first variable added to the list determines the data type to be used. For example, if you add an integer data type, any other variable added to the list must also be an integer data type.

The step type can be updated by clicking “The Minimum or Maximum of” option and selecting the option you want.

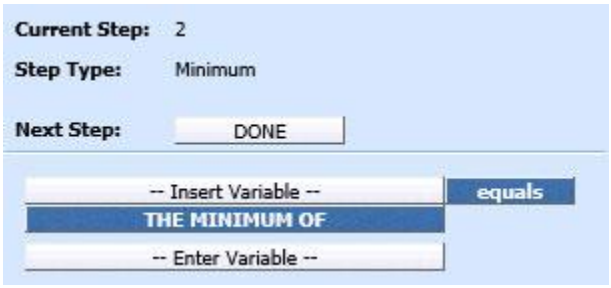


Figure 349 Minimum/Maximum Step

For example, if you wanted to determine the violation with the largest points, you would use the maximum step. A result variable is added to hold the largest point value from the list. The list of variables to be compared is added.

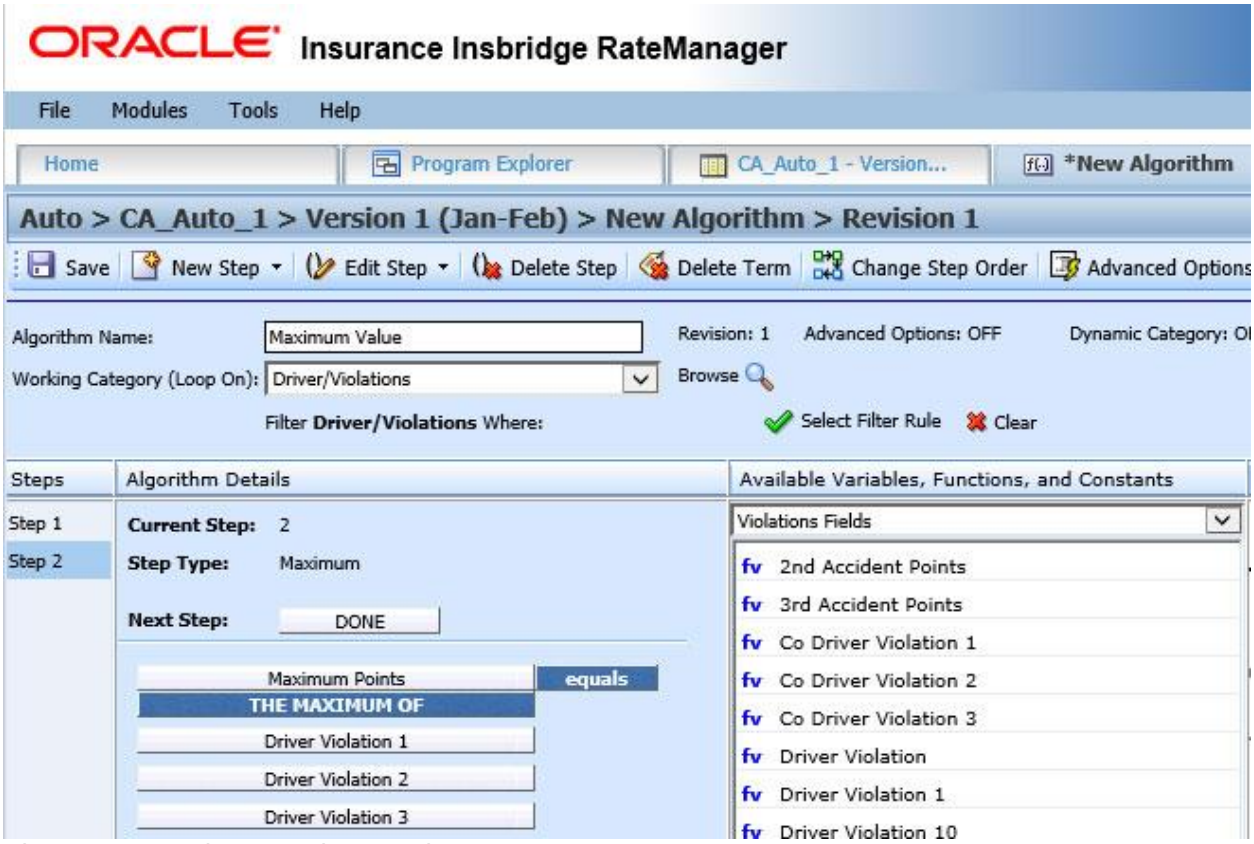


Figure 350 Creating a Maximum Value Step

When the step is run, the variables in the list are compared and the maximum value found in the list is returned.

Set Underwriting to Fail Step

A set underwriting to fail step is used in an algorithm when you want to halt rating because a policy does not meet an underwriting requirement.

For example, if, in the home product, you do not insure homes that are valued at more than $500,000, you could use an underwriting algorithm to check the value of the home and set underwriting to fail if it is greater than 500000. The algorithm could be placed at the beginning of the program sequence and, if it fails, processing time is not wasted calculating premiums for a policy that will not be written.

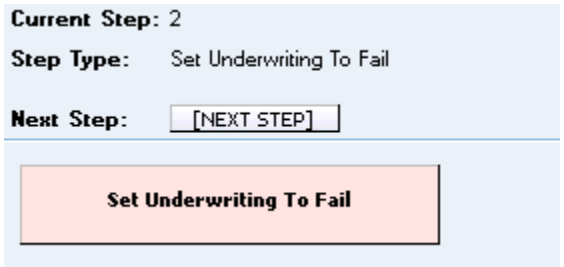


Figure 351 Setting Underwriting to Fail Step

If in a multi-occurring category, to stop rating, use EXIT LOOP as the next step. Otherwise, the remaining nodes will be processed.

Using Set Underwriting to Fail and Set Message

You can add a set message step after a set underwriting to fail to return a message. In this example, underwriting will fail if the effective date is within 10 days of the current system date.

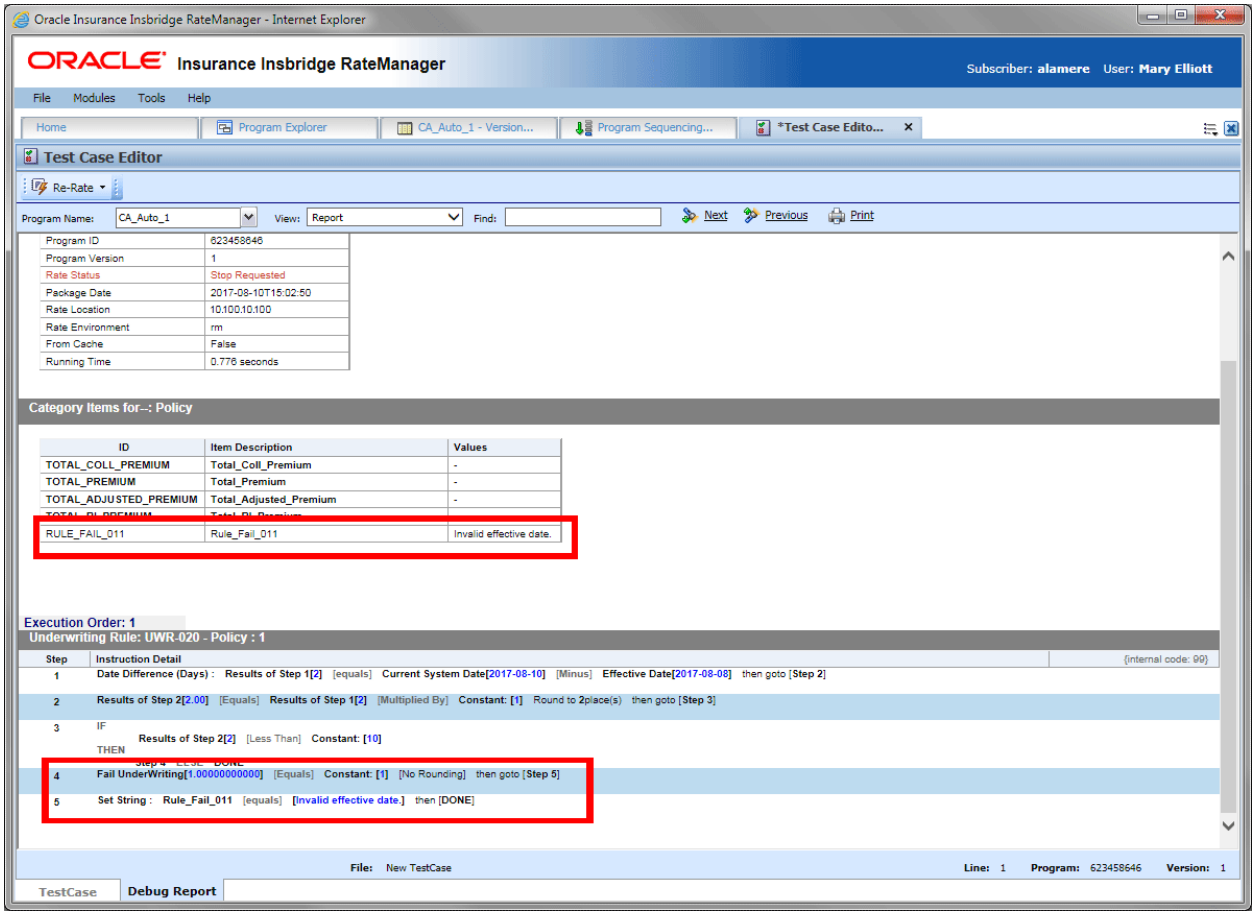


Figure 352 Set Underwriting to Fail and Set Message Steps

- Step 1: Determine the date difference between the effective date and the current system date.
- Step 2: Multiple the results of step 1 by 1.
- Step 3: If the result of step 2 is less than 10, go to step 4, set underwriting to fail, else done.
- Step 4: Set underwriting to fail.
- Step 5: Set Message where the result variable RULE_FAIL_001 contains the message “Invalid Effective date.”

Add the result for step 5 to the output mapping and the message is displayed if the underwriting rule fails. A message returned in the summary at the top of the debug report may be easier to locate than scrolling through the report looking for the rule that failed.

Set Principal Operator Variable

A set principal operator variable needs to be called prior to using selected assignment functions, **Assign All Vehs by Princ Op (Drvs NONEXCLUSIVE)** and **Assign Unassigned Vehs by Princ Op**. This step will set the principal operator indicator needed by the functions.

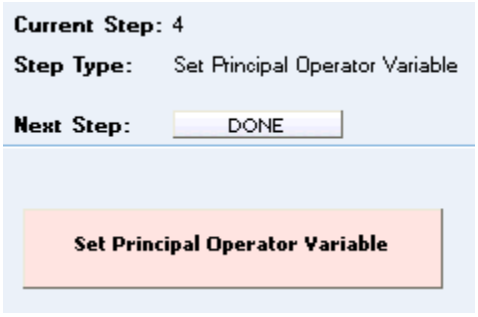


Figure 353 Set Principal Operator Variable

Re-calculate Vehicle Usage Step

The re-calculate vehicle usage step is used when you want change the vehicle usage option for a driver assignment scenario.

For example, at the beginning of a driver assignment scenario you need the vehicle usage option set to Vehicle Usage on Drivers, but at the end you need it set to Driver Usage on Vehicles. You would set the vehicle usage option for the scenario to Vehicle Usage on Drivers. When you need the vehicle usage option changed to Driver Usage on Vehicles, you would use the re-calculate vehicle usage step.

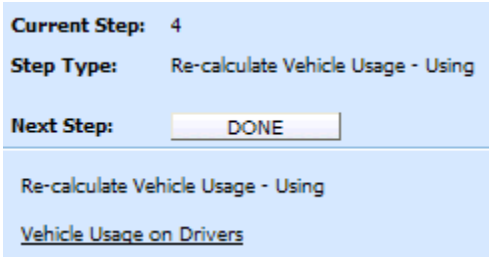


Figure 354 Re-calculate Vehicle Usage Step

RateManager provides four types of the re-calculate vehicle usage step. For more specific information on these options, see the driver assignment topic Vehicle Usage Options.

- **Vehicle Usage by Vehicle Order (InputId: 299):** Vehicle usage is designated as the actual vehicle a driver principally operates.
- **Vehicle Usage on Drivers:** Vehicle usage is designated as a percentage a driver operates each vehicle.
- **Driver Usage on Vehicles:** Vehicle usage is designated as a percentage a vehicle is operated by each driver.
- **Vehicle Usage by VehicleID (InputID: 299/368):** Vehicle usage is designated as the actual vehicle a driver principally operates, based on the vehicle number specified in the input file.

Working with Data Queries

A data query step is a step type that performs a query against the selected data source using the defined criteria. Unlike a table variable that runs when needed, a data query runs in the sequence listed.

Data query step types can be used in local algorithms where a dynamic category instance is used.

This step type is not available at the global or universal level, or to any other element, or if a dynamic category instance is not used. Data sources are added using the Query->Data Source step type.

Data query steps allow for data sources to be run multiple times during rating. When needed, you can change the criteria that the data source requires. The data source would remain the same.

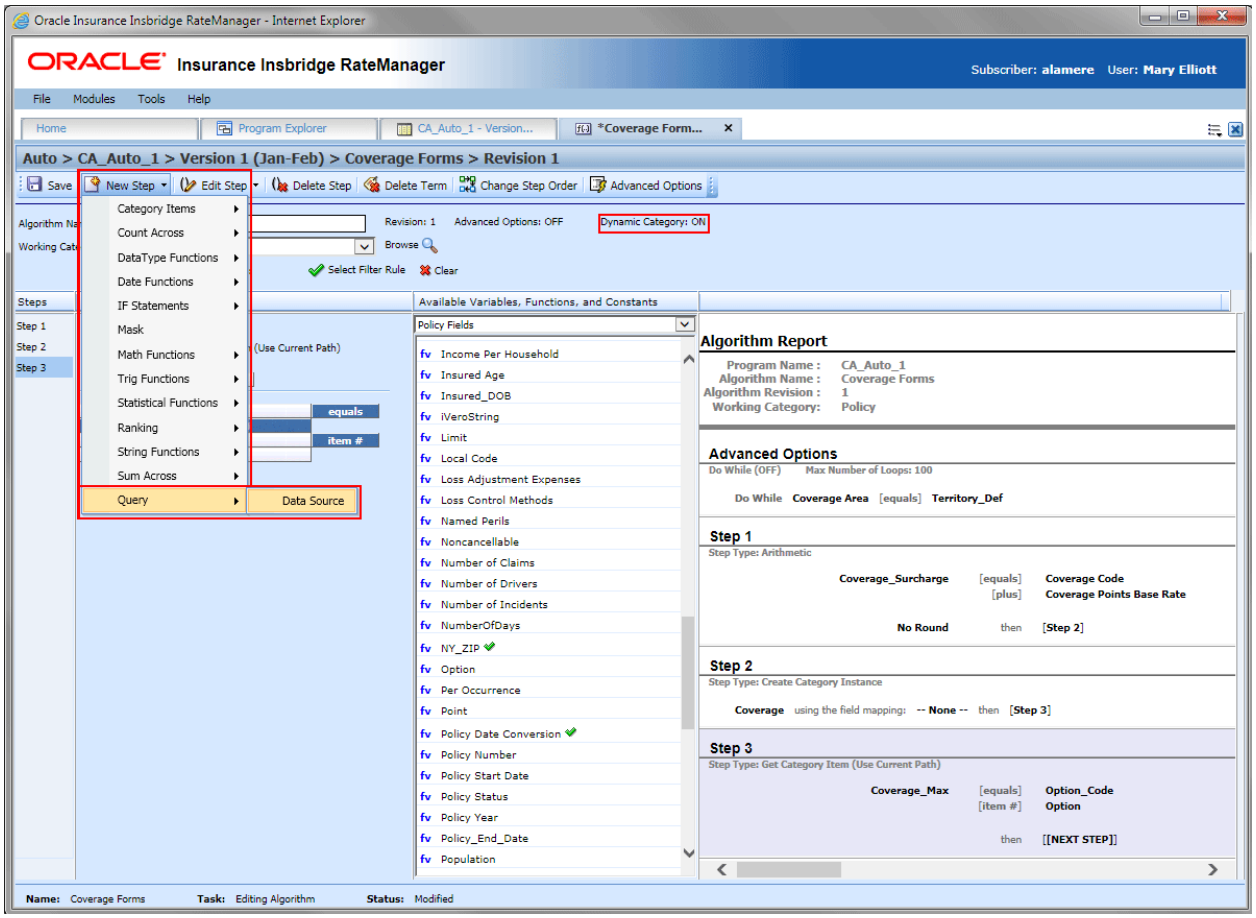


Figure 355 Entering a Data Source Query

To Enter a Data Source Query

Prior to creating a data source query, you should know the data source you want to use. You also need a result variable. The result variable and the data source data types must match.

The algorithm where you want to run a data source must be at the local level, and use a create category instance function. If the create category instance is removed, the algorithm cannot be saved until the data source is removed as well.

1. On either a new or existing local algorithm, select **New Step-> Query->Data Source**.

If the Query step type is not listed, verify that the Create Category Instance function has been used. If a Create Category Instance function has been used, save the algorithm and reload the screen.

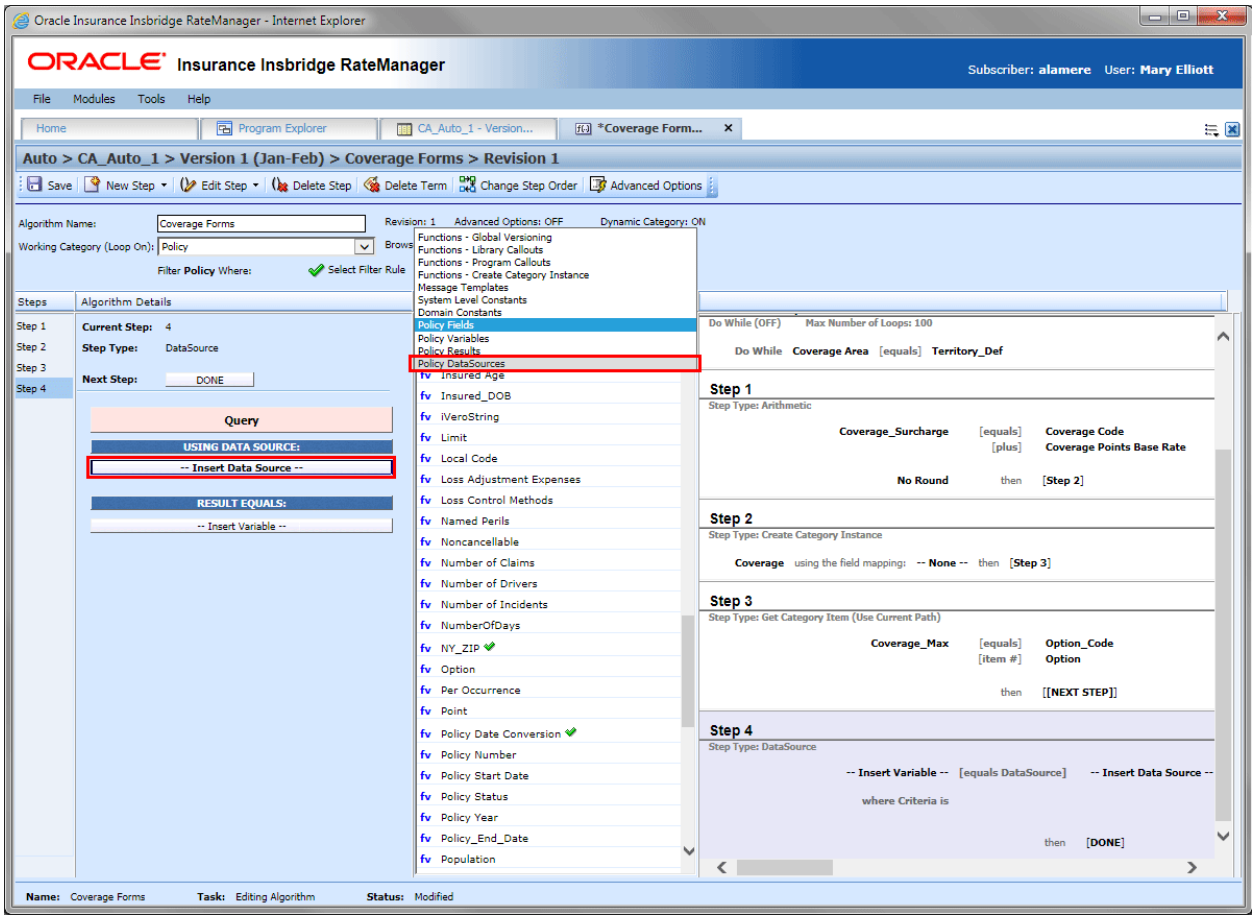


Figure 356 Selecting Data Sources in a Local Algorithm

2. Select **[Category] Datasources** from the Available Variables, Functions and Constants. The available data sources will be populated. Select the data source you want to use. This determines the criteria. The Datasources option depends on the category used.

3. Select the elements you want to populate to Data Query.

There are three fields:

- **Using Data Source:** This field holds the selected data source.
- **Result Equals:** This field holds a result variable. The data type must be the same as the data source.
- **Criteria:** This field is populated after you select the Using Data Source field. The criteria contain the data source table column headers. These are the headers that were brought in with the data source. They are column headers only. You do not have to match the criteria to the headers. You can select any criteria that you need. You must make a variable selection for each criterion. You cannot add or remove criteria from a data query.

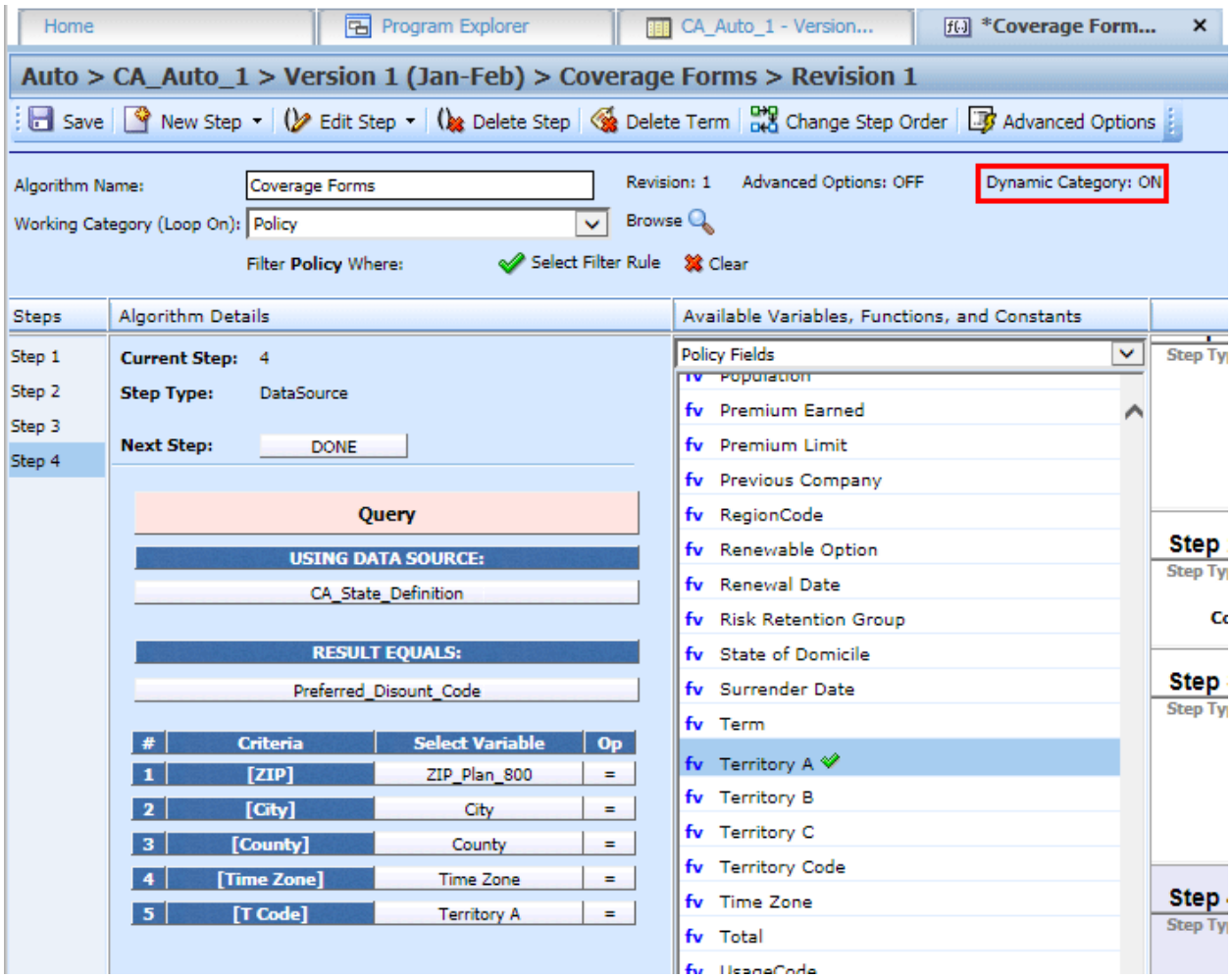


Figure 357 Completing a Data Query Algorithm

4. Once you are complete, finish the algorithm and save your work.

When the step is run, a query is performed on the data source that you specified, using the criteria you specified.

Data sources do have default values, if there is no match, the default value is returned.

NOTE: *Data Source Queries are only available in local algorithms where a dynamic category instance has been used.*

Chapter 18

CALCULATED VARIABLES

Calculated variables are used when a result cannot be derived from simple data mapping in a table. For example, if driver age is not passed as an input, but is a criteria needed in determining other factors, you would use a calculated variable to calculate driver age from fields of effective date and driver date of birth.

A calculated variable supports complex logic, such as IF Statements. Logic and functions within a calculated variable are created as steps. These steps are then added together to form a sequence of calculation.

Calculated variables can be created at three levels:

Universal: Universal variables are created and managed at the universal level and are project specific. Universal variables can be managed only from the Universal tab. Universal variables can be viewed on the Program Details tab but no edits, copies, or deletions can be done.

Global: Global variables are created and managed at the global level and are product specific. Global variables can be accessed from the Global tab or from the Program Details tab.

Local: Local variables are created and managed at the program version level and are specific to the program. Local variables can be accessed from the Program Details tab.

There are a few differences between universals, globals and locals. One difference between them is that universal variables are available to all programs in the project. Global variables are available to all programs within the product while program level variables are available only to the program where they were created. Another is that universals can use only universal elements, global variables can use universal and global elements. Local variables can use universal, global, or local elements.

Calculated variables can be template generated or created by you. Variables that are user created are open for editing, copying, creating new revisions and deletion if they are not locked.

Template generated calculated variables are locked and cannot be unlocked. No editing or deleting is allowed but they can be copied. This is to ensure the integrity of the template. New variables can be added to a template and new revisions can be created. These new variables are open for editing, copying or deletion, unless the program version is locked.

CALCULATED VARIABLE OPTIONS

- Creating a Calculated Variable
- Editing a Calculated Variable
- Copying a Calculated Variable
- Deleting a Calculated Variable
- Creating a Variable Revision Calculated
- Changing the Active Revision of a Calculated Variable
- Deleting a Calculated Variable Revision
- Creating a Custom Value
- Resolving Dependencies

Navigating to Calculated Variables

There are three paths to view or manage variables, either from the universals tab, globals tab or from the program details tab. The universal tab displays universal variables only. The global tab displays global variables only. You cannot view or manage local variables on the Global tab or the universal tab. On the program details tab, you can select to view either global variables or local variables. If you want to view and manage local variables, you must use the program details tab. The views and options are the same on either tab.

From either tab, when you select calculated variables in the program details folder, the current calculated variables are listed in the Folder Details area.

Calculated variables are not an option on the Program Explorer tab.

1. Modules→program explorer to open the program explorer tab.

2. Select the domain type→domain→project→product→folder→program→program version. You also can access calculated variables from the globals tab. Domain type→domain→project→product→globals or the universals tab via domain type→domain→project→universals.

3. Open the program version to get the Program Details tab. Calculated variables are available from the program details folders area on the left side of the tab.

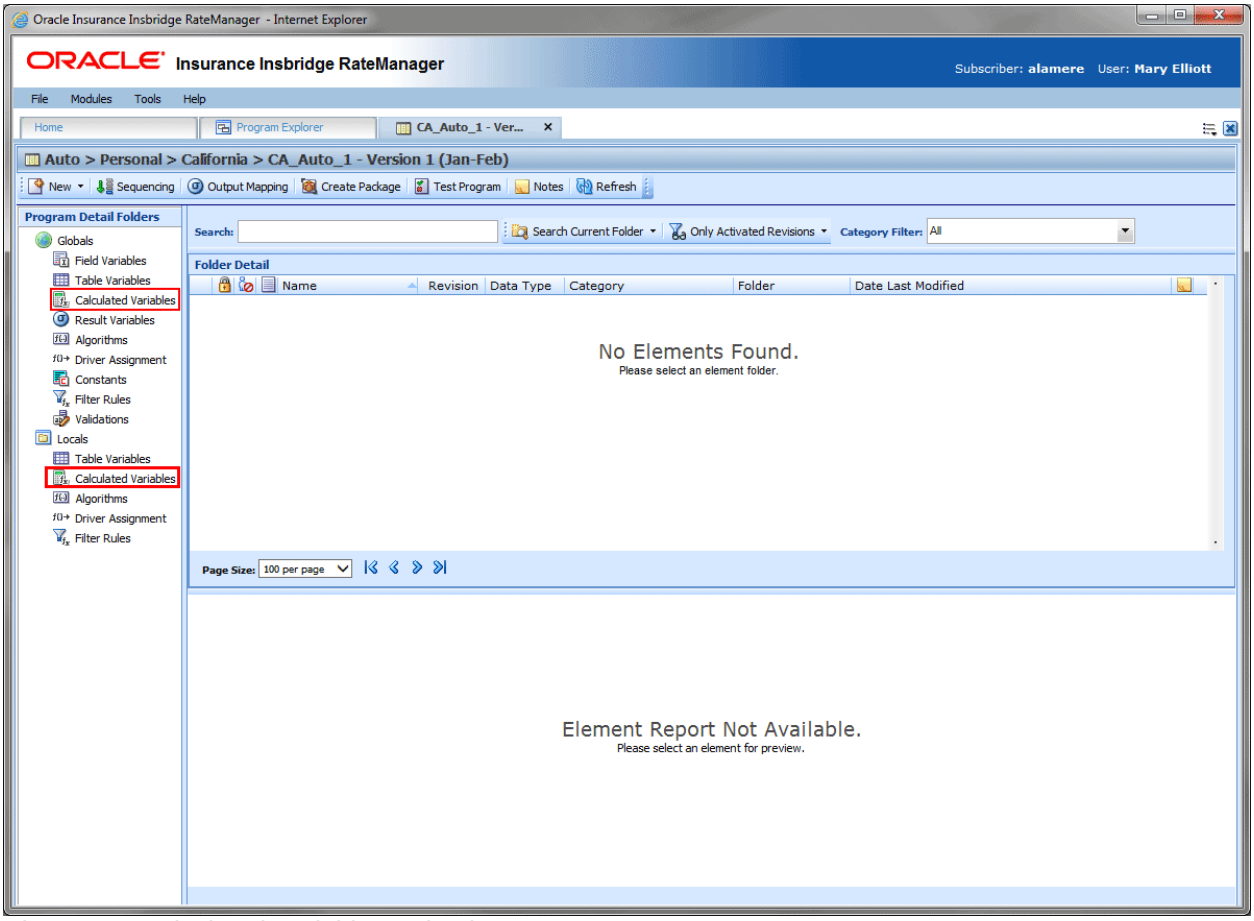


Figure 358 Calculated Variable Navigation

CALCULATED VARIABLE FUNCTIONALITY PROGRAM DETAILS TAB

Available functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any unlocked calculated variable.

- **Calculated Variable Functions:** The copy and delete functions can be performed on the Universal tab or Program Details tab using the right click menu. Reports also can be generated using the right click menu also.
- **Calculated Variable Revision Functions:** You can view, copy, and delete calculated variable revisions. Active revisions can be set on the Universal tab or the Program Details tab.
- **Calculated Variable Tab:** Creating a new calculated variable, editing an existing variable, or working with revisions is done on the Calculated Variable Details tab.

The functions you can perform depend on the status of the calculated variable. Locked calculated variables have fewer options. There are icons that can tell you at a glance what the status of the calculated variable is, if options have been applies and the active revision.

Icon	What it Indicates
	Calculated Variable. Universal calculated variables are project specific and can be created on the universals tab only. Global calculated variables are product specific and can be created at the global level, or from a program version. Local calculated variables are program specific and can be created at the program version level. The same options, new, edit, copy and delete are available on the universal, global or program details tab.
	Universal. Indicates that the calculated variable was created at the universal level. The calculated variable is grayed out and can be edited, copied or deleted from the universal tab only.
	Filter Rule. A filter rule has been applied to this variable
	Locked Variable. Locked calculated variables are not open for editing. A lock icon indicates that this calculated variable is locked by a program and is not open for copying, editing or deleting. If the calculated variable is locked because the program is locked, you must return to the program and unlock it. This unlocks the calculated variable and changes can be made.
	Template Lock. If the calculated variable is locked because it came from a template, you cannot unlock the variable.
	In-Use. Variables that are in use are currently being worked on by another user in the system or variables that you have open in another tab are not open for editing, copying, or deleting. The variable will be available when the user exits the variable.
	Active Revision. This variable is the current active revision.

Calculated Variable Screen Details

The screen details are the same for the Universal tab, Global tab and program details. For more on screen details descriptions, please see Program Details→Screen Details.

CALCULATED VARIABLE TAB

The Calculated Variable tab is available when you create a new calculated variable, edit an existing variable, or work with revisions. You also can be placed directly on the Calculated Variable tab from the *My Recently Opened Calculated Variable* snap on your Home tab.

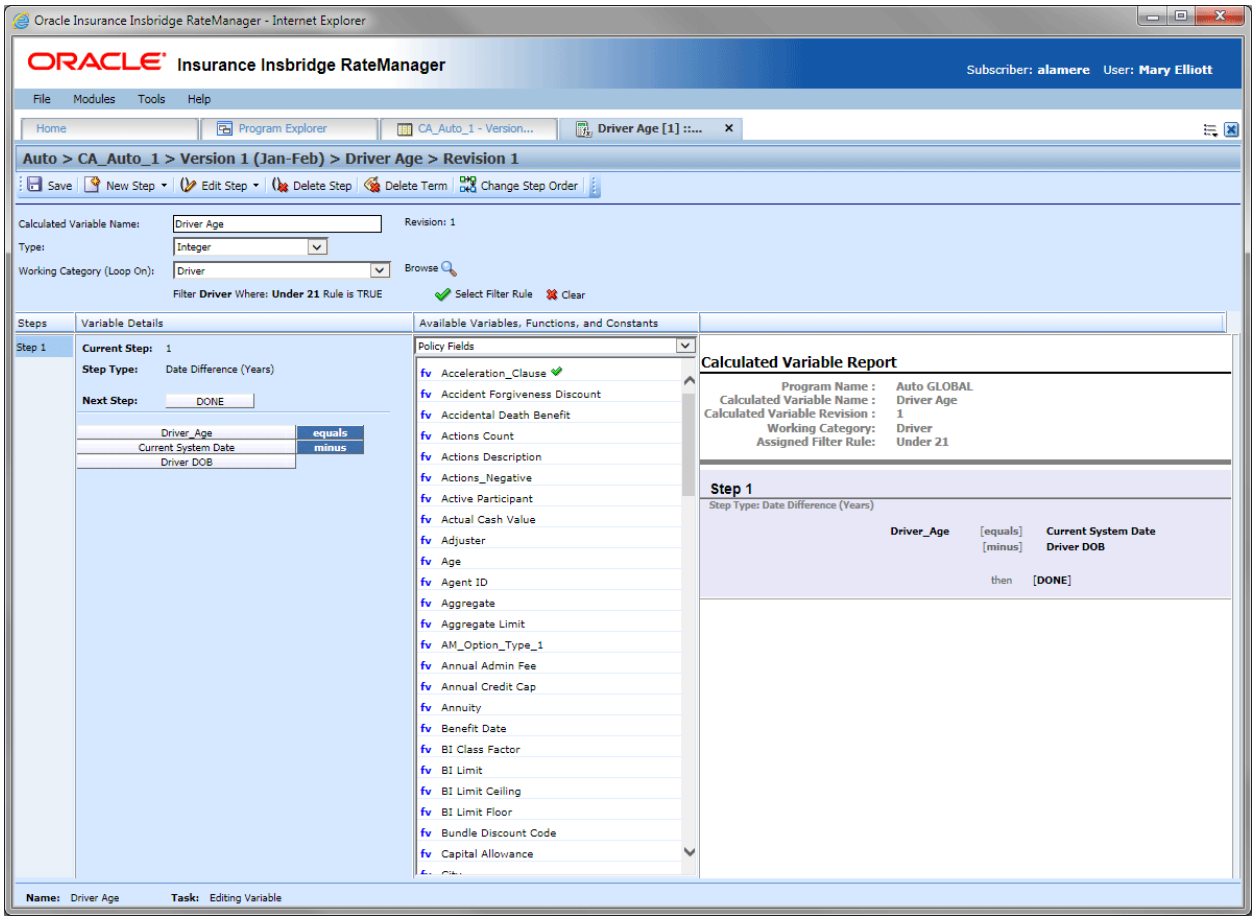


Figure 359 Calculated Variable Tab

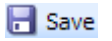
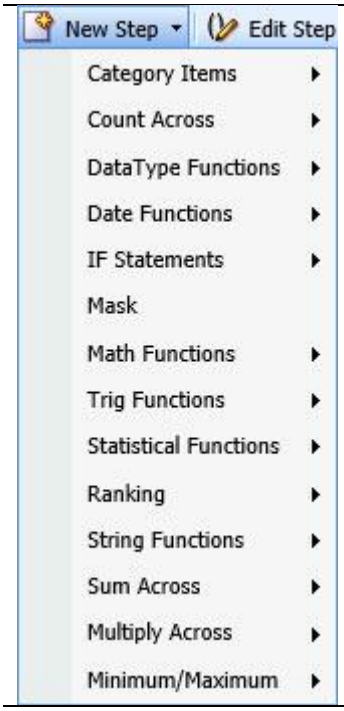
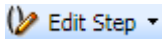
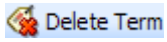
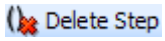
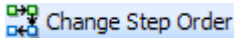
Calculated Variable Rules:

- The first step of a new calculated variable is always an arithmetic step. If you do not want an arithmetic step as your first step, add the step type you do want as Step 2. You can then delete the automatic arithmetic step.

- You cannot have a calculated variable without any steps. There must always be at least one step. You cannot delete the last remaining step.

CALCULATED VARIABLE FUNCTIONALITY

The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform on this tab.

	**Save:** Saves the calculated variable.
	**New Step:** Creates a new step in the calculated variable. For more information, see Working with Steps.
	**Edit Step:** Copies the selected step or pastes the steps on the clipboard in the selected step order.
	**Delete Term:** Deletes the currently selected term from a step in the calculated variable.
	**Delete Step:** Deletes the currently selected step. You cannot delete the last step of a calculated variable.
	**Change Step Order:** Allows users to change the order of the steps without using the drop and drag feature.

CALCULATED VARIABLE SCREEN DETAILS

There are four areas on the Calculated Variable Tab.

- **Variable Information:** The variable information area contains the name and defines the data type, and working category of the calculated variable.

- **Variable Details:** The details area is where you enter the steps that the calculated variable will perform.
- **Available Variables, Functions and Constants:** The list of the available variables, step types and constants that can be used in the Variable Details area.
- **Calculated Variable Report:** A listing of all the steps currently in the calculated variable.

Calculated Variable Information

The variable information area is where you define the calculated variable.

Naming standards can be applied to system elements. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the element and for case enforcement to be applied to any newly created element or any existing elements that are being edited. If a naming standard is in place, the pattern is displayed in the element name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented for all elements.

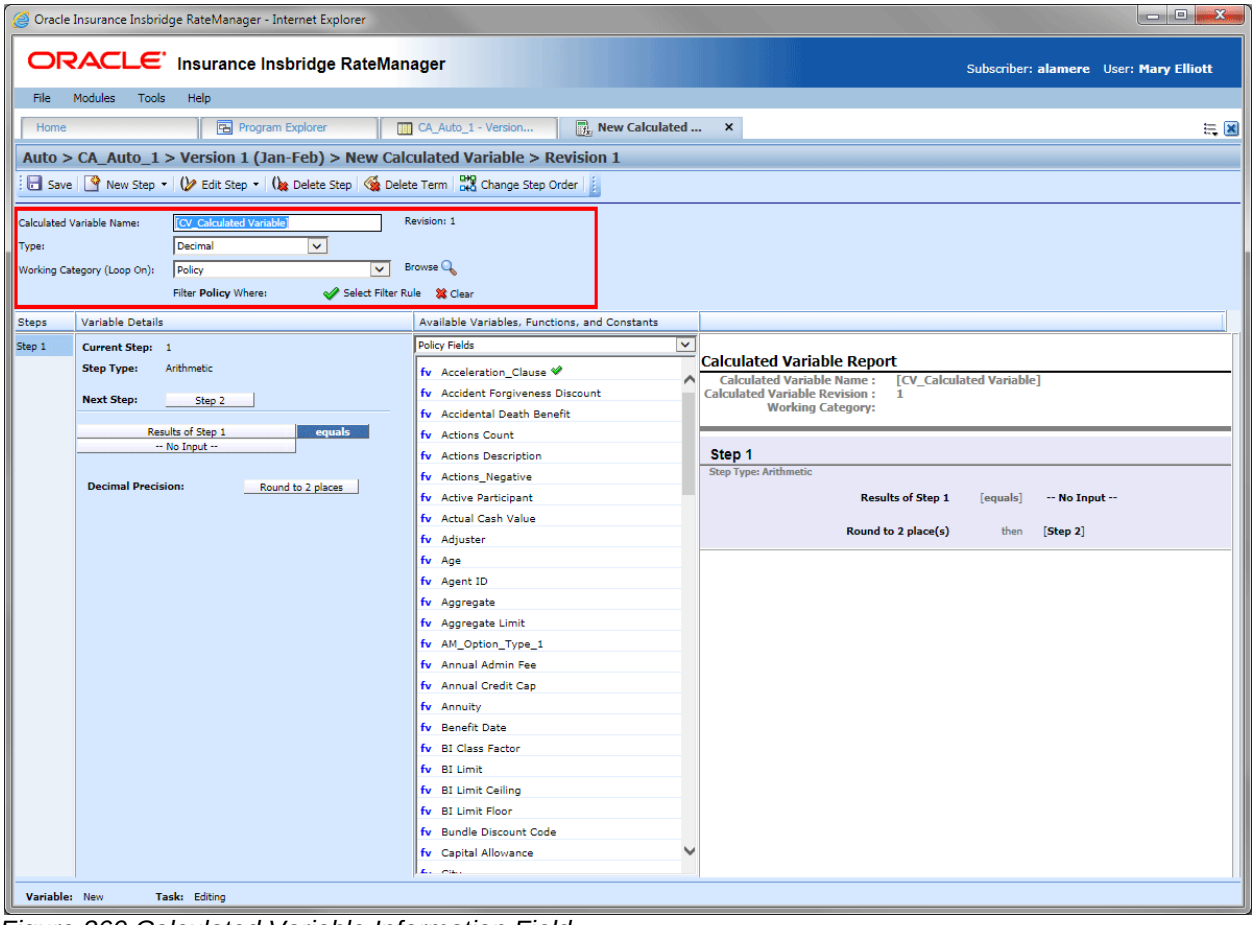


Figure 360 Calculated Variable Information Field

Screen Details

Calculated Variable Name: Entry for what you want the variable to be called. Alphanumeric characters are permitted with limited special characters (ex: &, *, ", +, @, etc.). Please see

Restrictions for more information. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

Type: Type of data associated with the variable. Options are:

- **Integer:** Used for whole numbers
- **String:** Used for any alpha numeric characters, including special characters
- **Decimal:** Used for numbers with decimal points
- **Date**: Used for dates

If disabled by the system administrator in the configuration file, the data type cannot be changed for any existing variables.

Working Category: Sets the category where this variable will be run. For example, if you have an auto test case with three vehicles, any variable set in the vehicle category will be run three times, once for each instance of the vehicle.

You can select the working category from the drop down menu or you can browse for a working category by clicking **BROWSE**. This brings up the Browse Category window. From here you can navigate to the Category you want. Once you have selected the category, click OK. The category will auto fill in the working category section.

The Working Category determines the options displayed in the Available Variables area. For example, if you select Policy as the Working Category, then the menu choices in the Available Variables area will be Constants, Step Variables, Policy Fields, Policy Variables and Policy Results. If you select Vehicle, the options will now include Vehicle Fields, Vehicle Variable and Vehicle Results along with Constants, Step Variables, Policy Fields, Policy Variables and Policy Results.

Filter Rule: Filter rules are assigned to table variables, calculated variables, or algorithms to determine if the calculated variable should be run. Click the green checkmark to add a filter rule. The rule can be removed by clicking the clear option. When a rule has been selected, the rule name is listed.

Calculated Variable Details

This area contains the steps that the calculated variable will perform.

A new calculated variable always begins with an arithmetic step as **Step 1**. If you do not want an arithmetic step as your first step, add the desired step and then delete the arithmetic step. To do so, highlight **Step 1** and click **Delete Step** in the menu bar at the top of the tab.

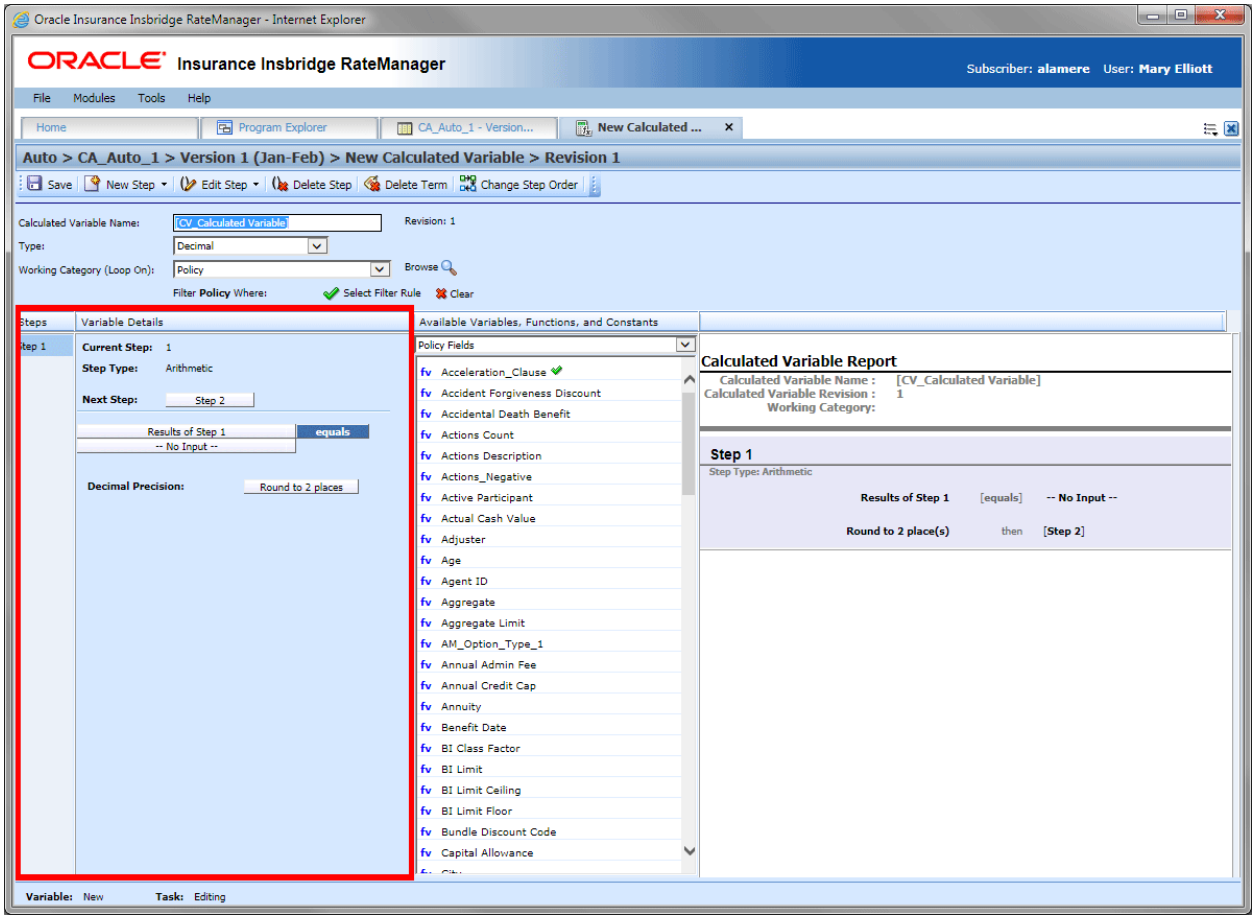


Figure 361 Calculated Variable Selection Criteria Field

Screen Details

Steps: Sequential order of steps created with drag-and-drop capabilities for moving a selected step up or down in the sequence.

Variable Details: Shows the selected step number, step type selected, next step to follow and fields/variables used for calculation. Rounding is set by clicking on the text box next to **Decimal Precision**. You can select a rounding precision from 0 to 10 digits.

Fields and variables will have a right click menu available; View Report, View Where Used and Custom Value.

Calculated Variable Available Variables, Functions and Constants

The Available Variables, Functions and Constants area contains a listing of available fields, variables, and results, sorted by category; system and domain constants; and message templates that can be used as criteria. For global variables, only other global variables or fields may be used. You cannot use any local variables in the creation of a global variable. A globe icon is used to indicate a global variable.

For Fields, you can add a new field variable without leaving the Edit Calculated Variable tab. Scroll to the bottom of the available variables list and select <New Field Variable>. A window is displayed that allows you to enter a new field variable.

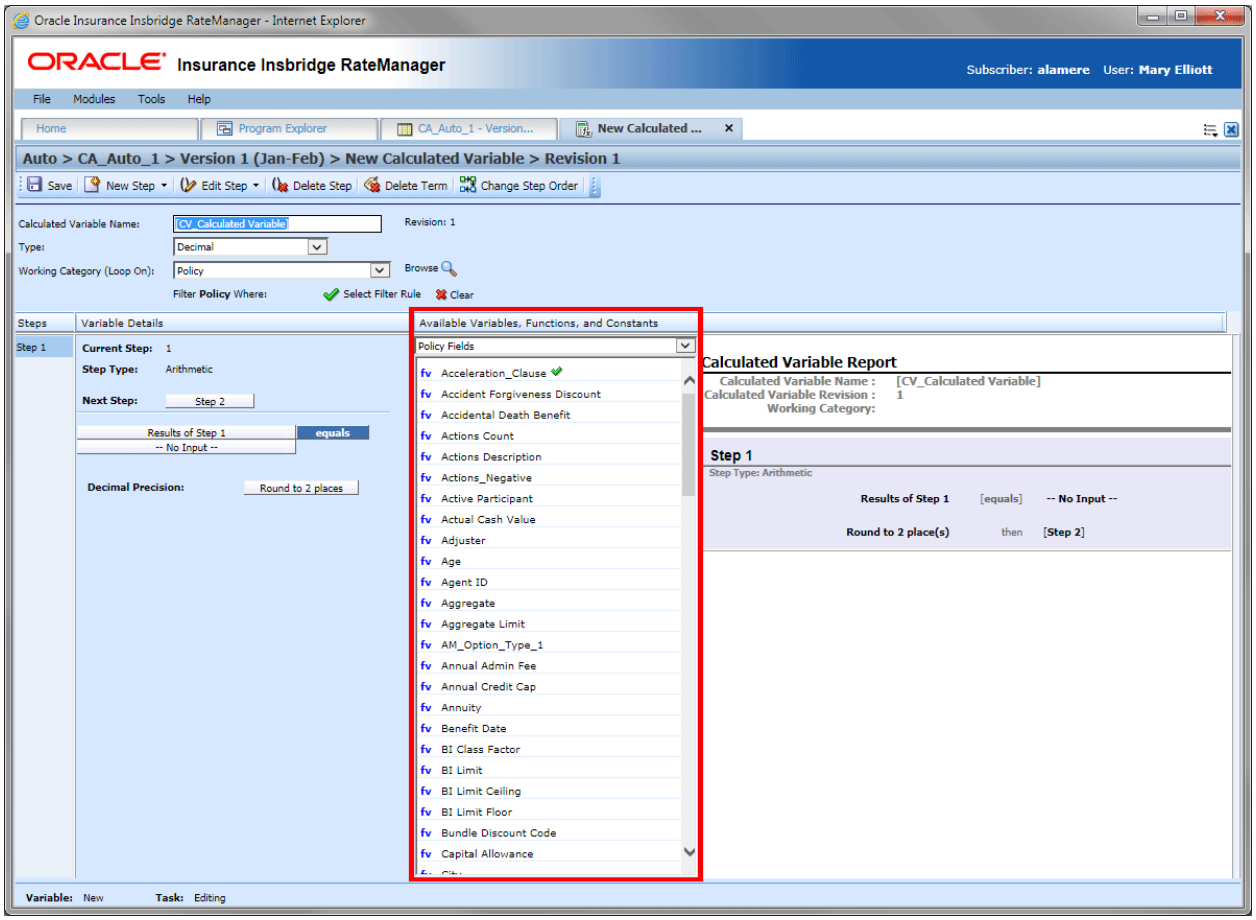


Figure 362 Available Variables, Functions and Constants Field

Screen Details

Menu: The menu is the top most field. Click the down arrow to select the category you want. Policy variables are available for all categories. Driver and Vehicle categories are available in the Driver-Vehicle category. Categories are determined in the Variable Information area.

Selection: The available variables, constants and step results are listed. Variables are based on the working category you select. Policy level fields variables and results are always listed. If you select another category, that categories fields, variables and results are listed.

Selection	Description
Message Templates	Message templates are prefabricated messages with variable placeholders that allow users to create standard messages using variable information that is generated at the time of execution.
System Level Constants	Pre-defined constants, including category instances and system date and time, and system wide user-defined constants. Or you can create a custom value for use in this program.
Domain Constants	User-defined constants found only in a selected product.
Step Variables	Results from the previously entered steps.
Policy Fields	Fields in the Policy category.
Policy Variables	Table and calculated variables in the Policy category.
Policy Results	Result variables in the Policy category.

Listing may also include other category fields, variables and results.

Calculated Variable Report

This area contains a detailed list of the steps you have entered. This information is view only. No actions can be performed in this area.

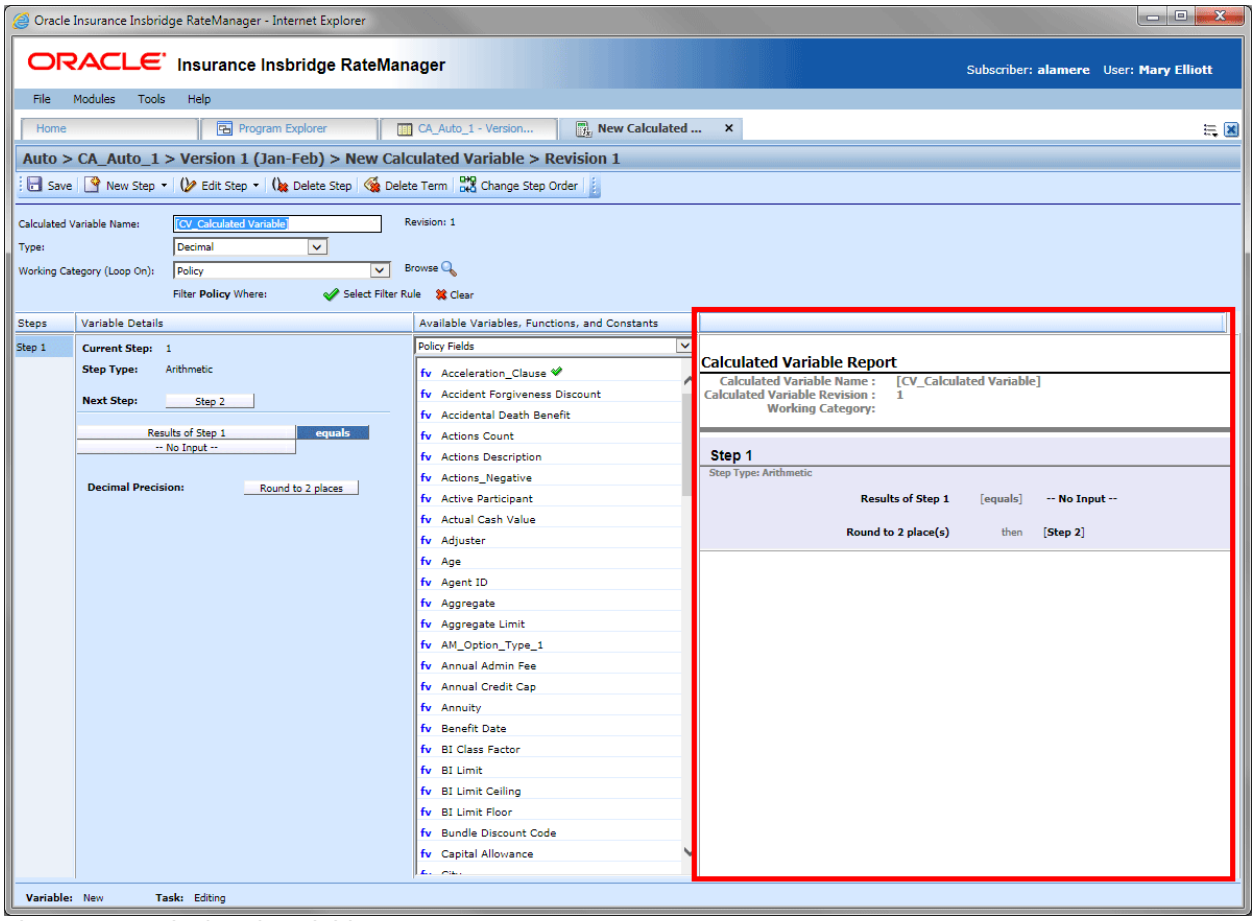


Figure 363 Calculated Variable Report

CREATING A CALCULATED VARIABLE

When you create a new calculated variable, it gets created in all versions of a program. Calculated variables can be added to both standard and templates programs. There is no limit to the number of variables that can be created for a program.

1. To create a universal variable, select the universal icon under the project where you want to create a variable on Program Explorer. To create a global or local table variable, enter the program details tab from the program where you want to create a variable. There are two options; select the global or local calculated variable option from Program Details Folder and click New from the menu bar or select New from the menu bar and then select calculated variable for the type of variable you would like to create, Global or Local. This opens the New Calculated Variable tab.

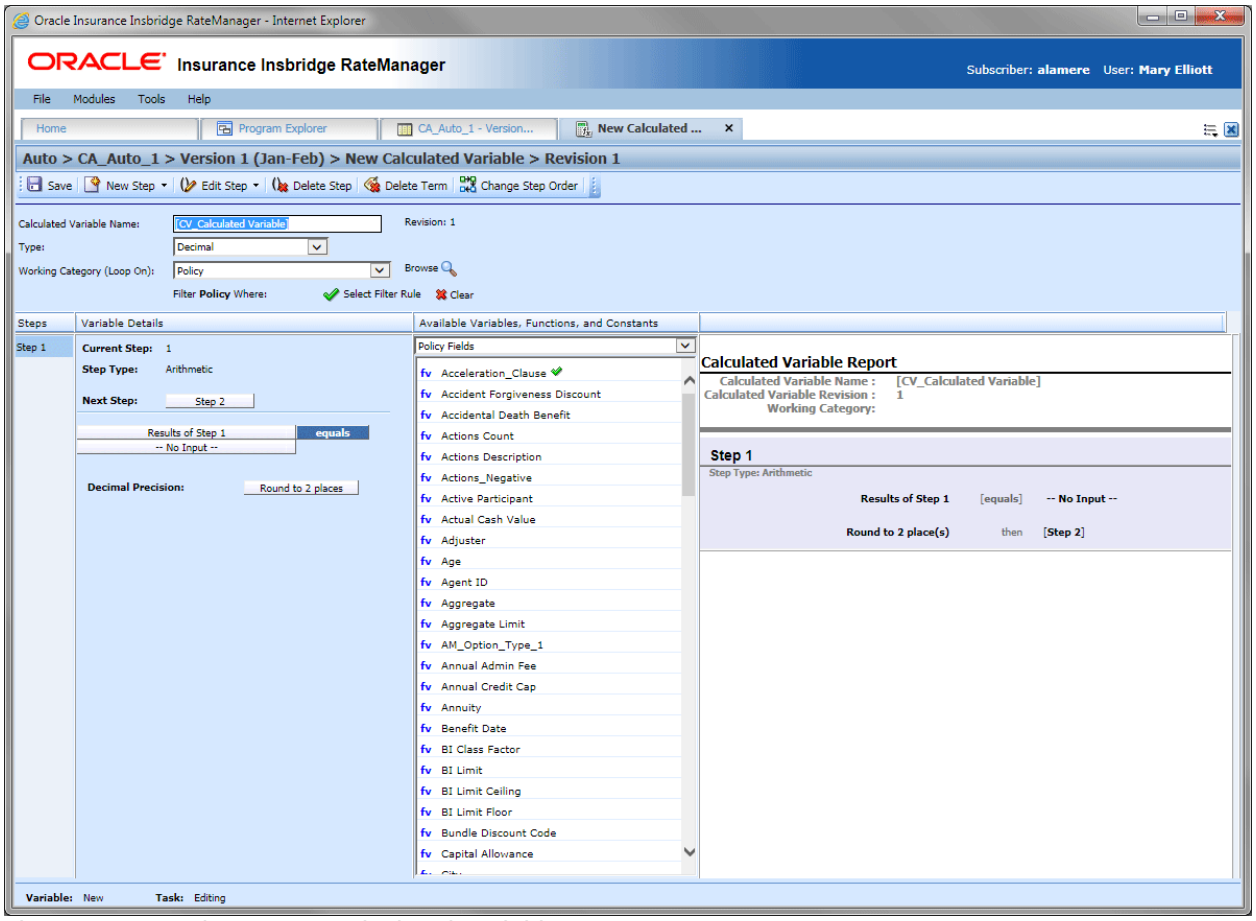


Figure 364 Creating a New Calculated Variable

2. Enter a **Name** for the calculated variable. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Select the **type** of data associated with the variable. Options are:
   - **Integer**
   - **String**
   - **Decimal**
   - **Date**

4. Choose a **Working Category**.

   You can select the working category from the menu or you can browse for a working category by clicking BROWSE to bring up the Browse Category window.

   From here you can navigate to the Category you want. Once you have selected the category, click **OK**. The category auto fills in the working category section.

   If this is not the screen you want, click **Cancel** to close the window and return to the previous screen.

NOTE: *Always save a calculated variable after defining the variable name, data type, and working category. The calculated variable will not be available to be used within a step until the variable has been saved.*

5. The first step is populated in the Variable Details area. This must be a mathematical step. If you do not want an arithmetic step as your first step, add the desired step and then delete the arithmetic step. To do so, highlight **Step 1** and click **Delete Step** in the menu bar at the top of the tab.

6. To add a new step, click **New Step** and select the type of step you want to add. For more information on the types of steps, see Working with Steps.

7. The screen refreshes with the new step parameters. Depending upon the step type, you can select the appropriate variables, functions, constants or fields. The values required are listed on the step.

   For example, if the step has an “Enter Variable” box, you must choose a variable from the menu in the Available Variables, Functions and Constants area. The options are populated underneath. Click the “Enter Variable” box and then double-click the corresponding variable you want. The box refreshes with your selected value.

8. Check the filter rule option to add a filter rule. See Adding Filter Rules for more information.

9. Continue entering steps until complete. When you are finished making your entries, click **Save**.

Adding a Variable or a Custom Value

A new field variable, or custom value can be created while entering or editing a calculated variable without leaving the Edit tab.

- To add a field **Variable**: See Adding a New Field from Another Area
- To add a **Custom Value**: See Custom Value

EDITING A CALCULATED VARIABLE

The **Edit Calculated Variable** tab is where you can change variable information such as name, data type, working category and steps. If the variable is locked, no changes are allowed.

Variables that are locked cannot be edited. Template generated variables that are using revision 1 are not open for editing. If you want to edit a template generated calculated variable, you must create a new revision. The new revision will be open for editing except for the name. The name cannot be changed.

NOTE: *If a change is made to a global calculated variable, the change affects all programs that use the global calculated variable. If a change only needs to be made to one particular program, create a local calculated variable in the program and have the program reference the local calculated variable instead of the global.*

To Edit a Calculated Variable

1. On the Program Details tab, select the **Calculated Variable** you want to edit.
2. Double click or right click and select **Edit**. The Edit Calculated Variable tab opens with the details of the variable.
3. Make any changes.

   To change the step order, you have two options:

   - **Option 1:** Select a step in the Steps column, drag it to a new location in the list and drop it.
   - **Option 2:** Select the step you want to move and click Change Step Order. A separate screen is displayed.
     - The current step position is displayed at the top of the screen. Select where you want the step to be placed.
     - Click OK. You will be returned to the edit calculated variable tab. Your step should be placed where you chose.

   To delete a step, select the step in the Steps column and then click Delete Step. You cannot delete the last step from a calculated variable.
4. Click **Save** to save your work.

WARNING: Category Updates to Revisions

Category updates should not be done to a revision. All revisions of a variable should be at the same category level. Changing the category for only some of the revisions of a variable is not supported.

CALCULATED VARIABLE COPY OPTIONS

Calculated variables may be copied in any program.

- **Local calculated variables** can be copied to the same program or to another program at the local level within the same product.
- **Global calculated variables** can be copied:
  - At the global level and are available to every program in the product
  - To the local program level
  - To another program within the same product at the local level
- **Universal calculated variables** can be copied at the universal level and are available to every program in the project.

The copy options depend on whether the calculated variable to be copied is at the universal level, global level or the local level.

Local Copies

Local copies can be done to the local level of any program in the product. When copying a local calculated variable to a local level in another program:

- Local copies are done at the product level. The Destination Program name is listed first and the folder is in parenthesis.
- Local to local copies check for dependencies when the copy is made:
  - When performing a local to local copy, dependencies such as the usage of other local elements are checked. If there are local elements used in the element being copied that are not in the Destination Program, then those elements are copied also.
  - For example, the Option AB Fee algorithm, a local variable, uses table variables Option A and Option B, also local variables. When Option AB Fee is copied from program Maine to program Kansas, the algorithm is copied and so is the table variables that are being used.
- Local to local copies do not require a name change:
  - When copying to the local level of a different program, the table variable name cannot be changed. The New Name is grayed out. This creates an identical variable in the other program and reduces the need for editing. If a change is made to the original variable and the variable is copied for a second time, the copy is created as a new revision in the Destination Program.
  - When copying at the local level in the same program, the name must be changed.
- Copies made in the current program must have a new name.

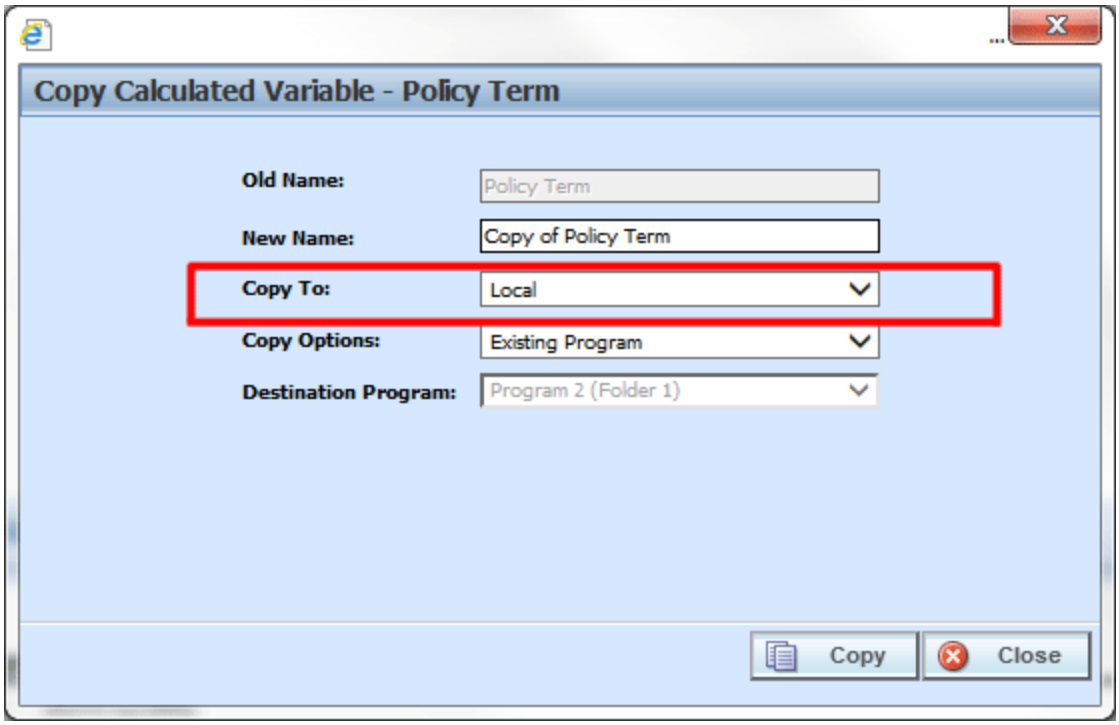


Figure 365 Copying Calculated Variable at the Local Level

Global Copies

When a global calculated variable is copied to a local level or when a local calculated variable is copied, there are two copy options:

- **Existing Program:** Copy the variable into the existing program under another name.
- **Other Program:** Copy the variable into another program within the same product.

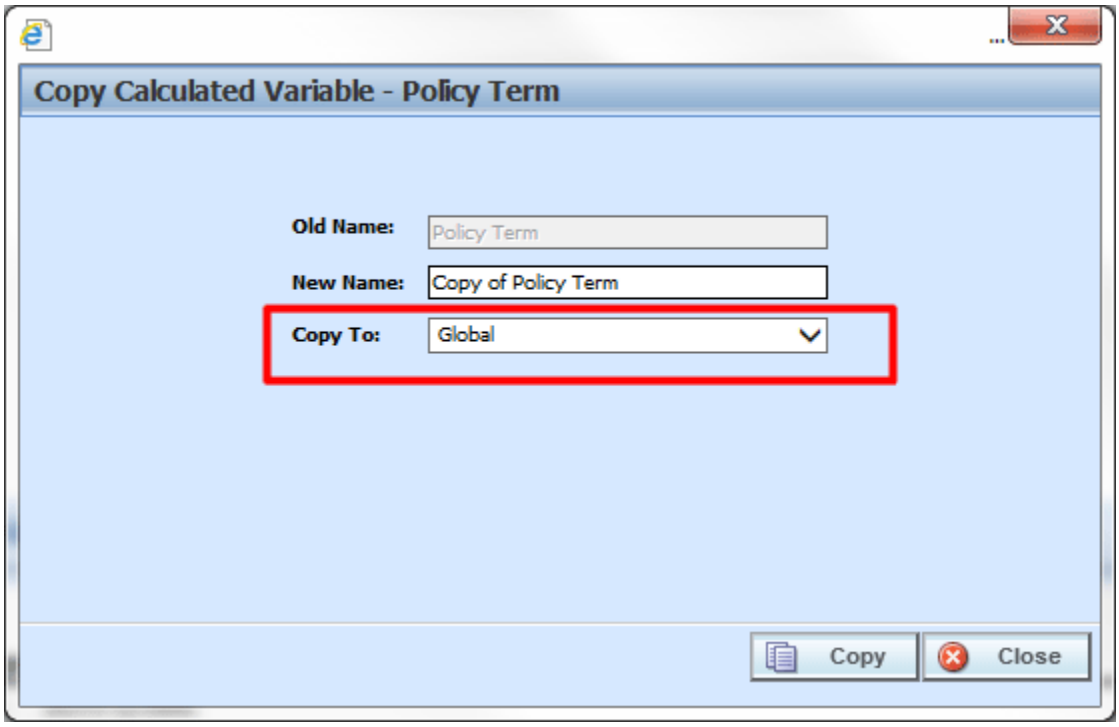


Figure 366 Copying Calculated Variable at the Global Level

- Global copies are done at the product level. The Destination Program name is listed first and the folder name is in parenthesis.
- The New Name must be different from the Old Name.

Universal Copies

When copying a universal variable to a local level:

- Universal copies to a local level are project wide. Information has been added to help you select the Destination Program where you want to copy the element to. The Destination Program name is listed first, the folder name is in parenthesis and the product name is last, also in parenthesis.
- Duplicate naming is not allowed. The New Name must be different from the Old Name.
- Product level naming standard are not enforced. If the product has a naming standard, copies made from the universal level are not required to have the naming standard applied when copying. If the variable is edited any time after the initial copy, the naming standard is enforced.
- Universal permissions are required to make copies at the universal level.

COPYING A CALCULATED VARIABLE

11. Select the calculated variable you want to copy and right click. Select **Copy** from the menu.

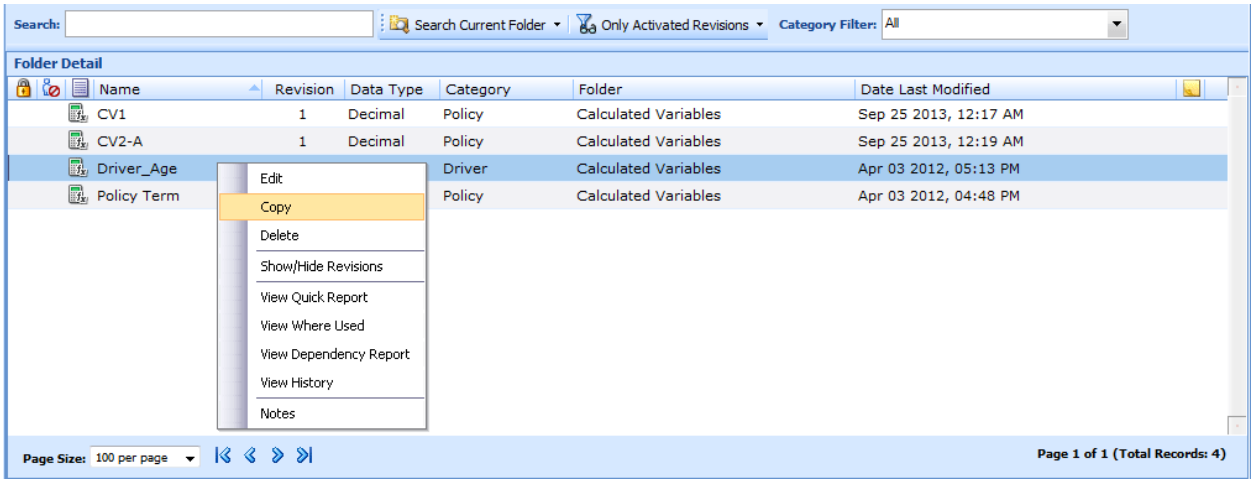


Figure 367 Copy Variable Menu

12. A separate window is displayed.

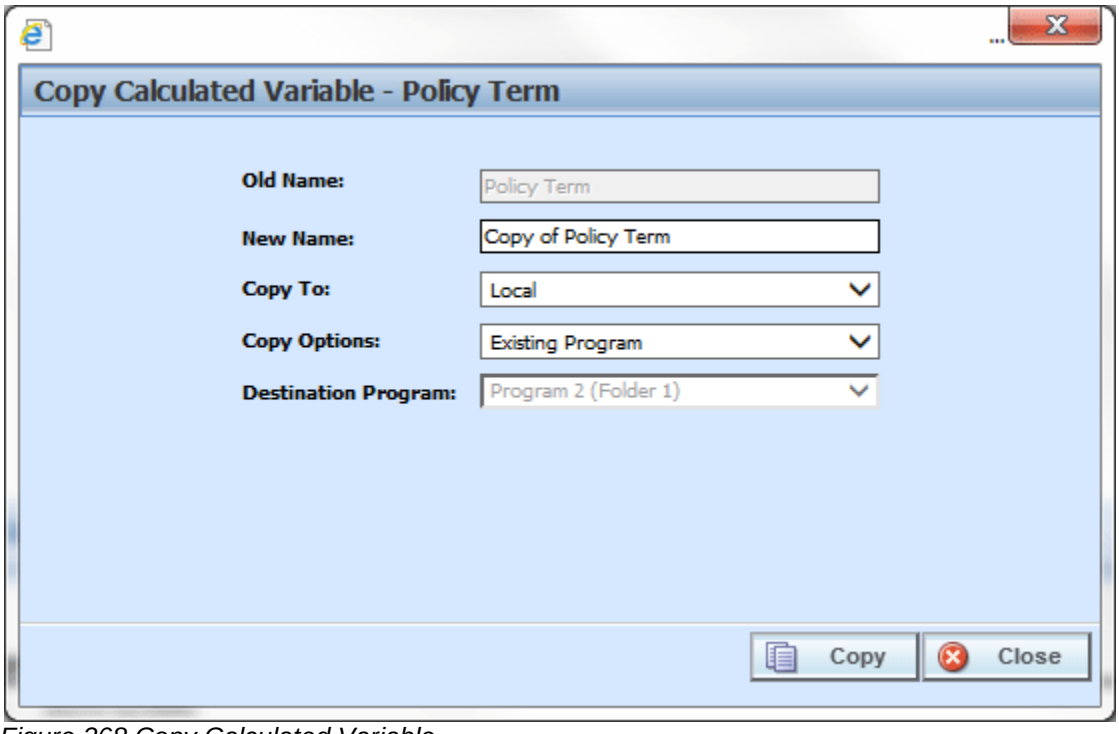


Figure 368 Copy Calculated Variable

13. Select the **Copy Options**. It is recommended that you change the copy options first. Changing options may reset the name field.

14. If this is a global variable, select if this is a local or global copy. This option is not present on local calculated variables.

15. The copy options determine the Destination Program selections. If you are copying to the current program, you cannot choose the destination program. If you are copying to another program, select the **Destination Program** from the drop down menu.

16. Enter in a **New Name**. The name can be entered first. It is recommended that you change the copy options first. Changing options may reset the name field.

17. Click **Copy** to copy the variable. Dependencies will be searched. If there are any dependencies that need to be resolved, a window is displayed. See Resolving Dependencies.

18. If the copy was successful, you will be returned to the previous screen. Your variable has been copied. The last updated date is the date the copy was made.

DELETING A CALCULATED VARIABLE

When a variable is no longer needed, it can either be left in RateManager or deleted. When you delete a variable, it gets deleted from all versions of the program. Also, all revisions of that variable get deleted. If you are uncertain as to whether the variable will be used again, simply leave it in RateManager. This will save time if the variable is needed and won't slow down rating.

To Delete a Variable

1. Select the variable you want to delete. Right click it and select **Delete**.

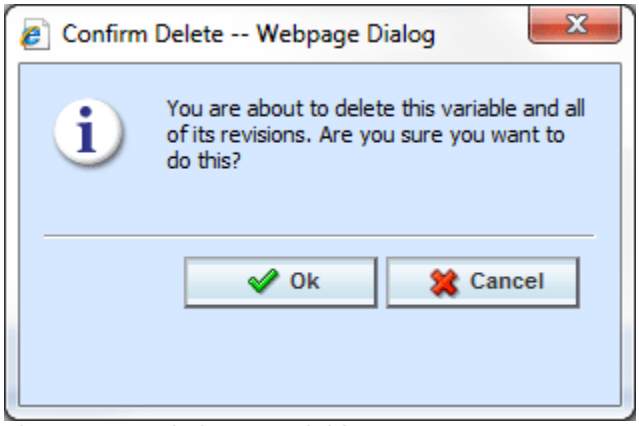


Figure 369 Deleting a Variable

2. You will be asked to confirm the deletion of the variable. Click **Ok** to delete the variable. Click **Cancel** to return to the tab without affecting a change.

If you try to delete a variable that is still being used by an algorithm, a result group or another variable, you will receive an error. Variables that are locked or that are template generated cannot be deleted.

NOTE: *Variables that are in use cannot be deleted.*

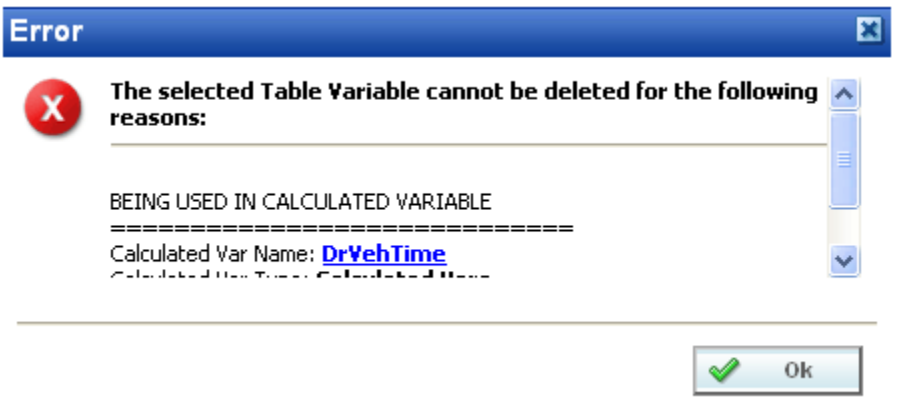


Figure 370 Error Message for Deleting a Variable

RESOLVING DEPENDENCIES

Dependencies may arise when you copy a variable, algorithm or driver assignment. Some dependencies may involve naming issues or may require you to create new variable. All dependencies will be listed. The options that you will have to choose from will not create further dependencies. For driver assignment, only available driver assignment functions will be shown.

The action column can contain three options:

- **Create**: This creates a new variable.
- **No-Action**: The variable in the Target Program Var field will be used.
- **Replace**: A new variable has been selected.

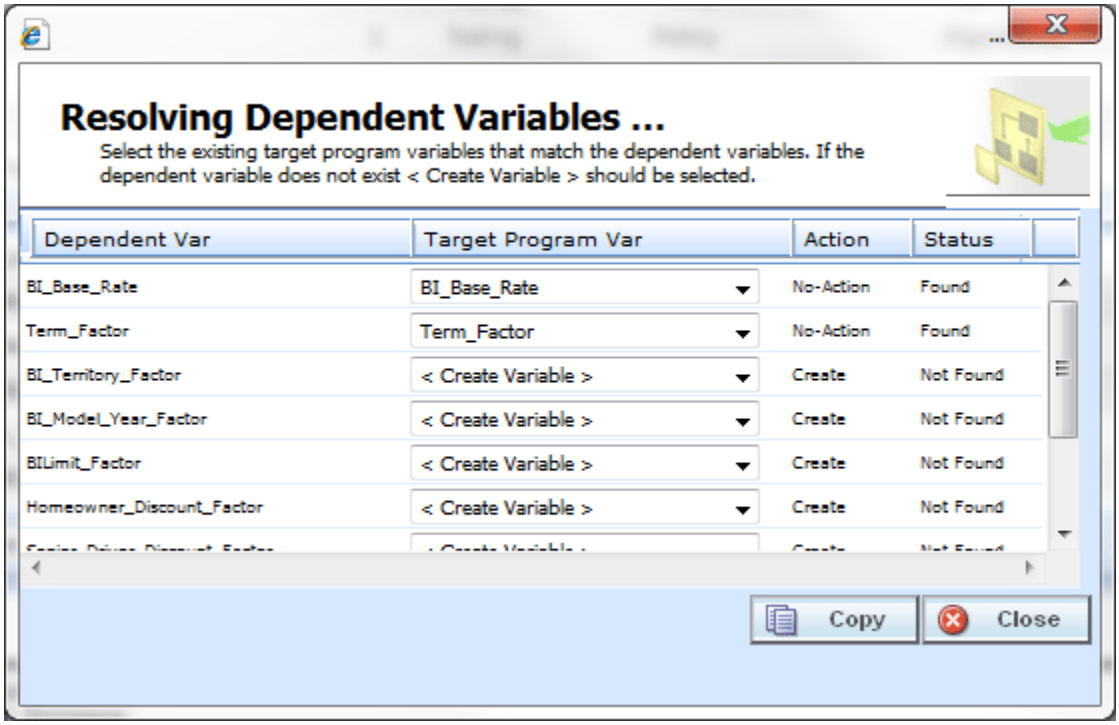


Figure 371 Resolving Dependent Variables

1. Select the **Target Program Var** you want from the menu. The menu will only contain one option if only one option can be taken.
2. Click **Copy**. The dependencies will be resolved and you will be returned to the previous screen. Your variable will have been copied. The last updated date will be the date the copy was made.

Chapter 19

ALGORITHMS

Algorithms in RateManager are similar to calculated variables in that they allow multiple steps to be used in determining a value. Algorithms are different from calculated variables in that you can select for an algorithm to be run when you want it to run while calculated variables are run only when needed. Algorithms are most commonly used in determining a premium or checking an underwriting condition. RateManager can create two different kinds of algorithms, Rating Algorithms and Underwriting Algorithms.

- **Rating Algorithms** are mainly used for determining premiums. However, since algorithms can be sequenced to run in a specific order, they can be used to create a result for differences in limits being passed versus rated, tier assignments, and so on.

- **Underwriting Algorithms** are used to ensure underwriting rules have been met, prior to rating a policy, which saves valuable execution time. Underwriting algorithms are flexible and can be implemented separately per tier or combined using a tiered structure. Messages can be set to output as a result, with or without the policy being set to fail on the rule. Underwriting can be designed to fail on the first rule or fail on all that may be applicable, allowing you to see all of the messages associated with that applicant.

There are differences in functionality between rating algorithms and underwriting algorithms.

- **Rating Algorithms** can have **advanced algorithm looping** added. This option allows for a rating algorithm to be run a specified number of times until a condition is met. This option is not available in underwriting algorithms.

- **Underwriting Algorithms** have one additional step type available; **Set Underwriting to Fail**. During run time, if the policy hits this step in the underwriting algorithm, rating will stop for that specific policy. The remainder of the current algorithm will be processed, but no other algorithms or premiums will be calculated.

  The advantage of using the **Set Underwriting to Fail** step is when there are some “hard and fast” underwriting rules that, if failed, will cause the policy to not be written under any circumstances. You have the ability to place these underwriting rules run at the very beginning of the program. If the rule does fail, processing time is not wasted in calculating premiums for a policy that would not be written.

Universal, Global and Local Algorithms

Rating Algorithms and Underwriting Algorithms can be created and managed at the universal level, global level and at the local level.

There are a few differences between universals, globals and locals. One difference between them is that universal variables are available to all programs in the project. Global variables are available to all programs within the product while program level variables are available only to the program where they were created. Another is that universals can use only universal elements, global variables can use universal and global elements. Local variables can use universal, global, or local elements.

Universal: Universal variables are created and managed at the universal level and are project specific. Universal variables can be managed only from the Universal tab. Universal variables can be viewed on the Program Details tab but no edits, copies, or deletions can be done.

Global: Global variables are created and managed at the global level and are product specific. Global variables can be accessed from the Global tab or from the Program Details tab.

Local: Local variables are created and managed at the program version level and are specific to the program. Local variables can be accessed from the Program Details tab.

Algorithms can be template generated or created by you. Algorithms that are user created are open for editing, copying, creating new revisions and deletion if they are not locked.

Template generated algorithms are locked and cannot be unlocked. No editing or deleting is allowed but they can be copied. This is to ensure the integrity of the template. New algorithms can be added to a template and new revisions can be created. These new algorithms are open for editing, copying or deletion, unless the program version is locked.

Results from Algorithms

Algorithms should be used to set the value of result variables. Result variables are a special type of global variable that are available across the project or the product depending on where the result was created.

Result variables can be used in local algorithms, global algorithms, and universal algorithms. Result variables are typically premium totals and underwriting messages that are specified as output in the Result Mapping. Result variables are not instantiated, so if their value is not set by an algorithm, it remains null. Two reasons to use algorithms set result variables include:

Example 1: Multiple result variables from a single algorithm

BI Premium can be set and Monthly BI Premium can be set, which would divide the BI Premium by Term Duration.

Example 2: Multiple algorithms setting a single result variable

BI Premium can be set by both Private Passenger BI Algorithm and Motorcycle BI Algorithm, depending on vehicle type.

Algorithm Rules

- The first step of a new algorithm is always an arithmetic step. If you do not want an arithmetic step as your first step, add the step type you do want as Step 2. You can then delete the automatic arithmetic step.
- You cannot have an algorithm without any steps. There must always be at least one step. You will not be able to delete the last remaining step.

ALGORITHM OPTIONS

The **Program Details tab** and the **Globals tab** shows all algorithms, either rating or underwriting, that have been created for a program. In addition, the revision number, type and working category of the algorithms will be displayed. From this tab you will be able to:

- Creating a new Algorithm
- Using an Advanced Algorithm Looping
- Editing an algorithm
- Copying an Algorithm
- Deleting an Algorithm
- Creating a new Algorithm Revision
- Changing the Active Revision of an Algorithm
- Deleting an Algorithm Revision

Navigating to Algorithms

There are three paths to view or manage variables, either from the Universals tab, Globals tab or from the Program Details tab. The universal tab displays universal variables only. The global tab displays global variables only. You cannot view or manage local variables on the Global tab or the universal tab. On the Program Details tab, you can select to view either global variables or local variables. If you want to view and manage local variables, you must use the Program Details tab. The views and options are the same on either tab.

From either tab, when you select Algorithms in the Program Details Folder, the current algorithms will be listed in the Folder Details area.

Algorithms will not be an option on the Program Explorer tab.

1. Modules→Program Explorer to open the Program Explorer tab.

2. Select the domain type→domain→project→product→folder→program→program version. You also can access algorithms from the globals tab. Domain type→domain→project→product→ globals or the universals tab via domain type→domain→project→universals.

3. Open the program version to get the Program Details tab. Algorithms are available from the program details folders area on the left side of the tab.

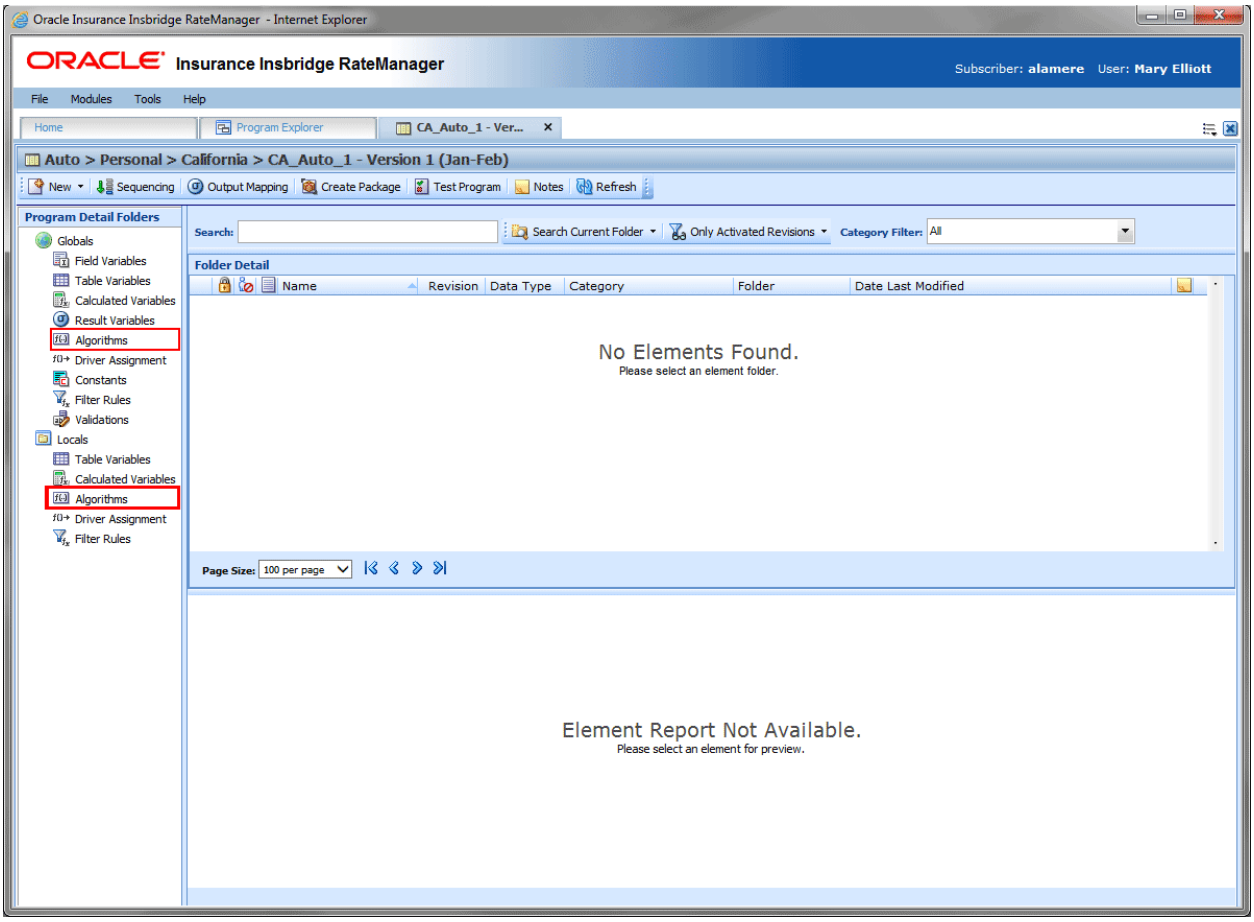


Figure 372 Algorithms Navigation

ALGORITHM FUNCTIONALITY PROGRAM DETAILS

Available functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any unlocked algorithm.

- **Algorithm Functions:** The copy and delete functions can be performed on the Program Details tab using a right click menu. Reports also can be generated using the right click menu on the Program Details tab.
- **Algorithm Revision Functions:** You can view, copy, and delete algorithm revisions. Active revisions can be set on the Program Details tab as well.
- **Algorithm Tab:** When you create a new algorithm, edit an existing algorithm, or work with revisions, you will be placed on the Algorithm tab.

The functions you can perform will depend on the status of the algorithm. Locked algorithms have fewer options. There are icons that can tell you at a glance what the status of the algorithm is.

Icon	What it Indicates
	Algorithm. Universal algorithms are project specific and can be created on the universal tab only. Global algorithms are product specific and can be created at the global level, or from a program version. Local algorithms are program specific and can be created at the program version level. The same options, new, edit, copy and delete are available on universal, global or program details tab.
	Universal. Indicates that the algorithm was created at the universal level. The algorithm is grayed out on the Program Details tab and can be edited, copied or deleted from the universal tab only.
	Locked Algorithm. Locked Algorithms are not open for editing. A lock icon indicates that this algorithm is locked by a program and is not open for copying, editing or deleting. If the algorithm is locked because the program is locked, you must return to the program and unlock it. This will unlock the algorithm and changes can be made.
	Template Lock. If the algorithm is locked because it came from a template, you will not be able to unlock the algorithm. You can create a new algorithm if necessary.
	In-Use. Algorithms that are in use are currently being worked on by another user in the system or algorithms that you have open in another tab are not open for editing, copying, or deleting. The algorithm will be available when the user exits the variable.
	Advanced Option. This algorithm has had advanced algorithm looping applied to it.
	Dynamic Category Instance. Indicates that the algorithm has a Create Category Instance function. Local algorithms only.
	Active Revision. This algorithm is the current active revision.

Algorithm Screen Details

The screen details are the same for the Universal tab, Global tab and program details. For more on screen details descriptions please see Program Details→Screen Details.

ALGORITHM TAB

The Algorithm tab is available when you create a new algorithm, edit an existing algorithm, or work with revisions. You also can be placed directly on the Algorithm tab from one of the six *My Recently Opened Algorithms* snaps on your Home tab.

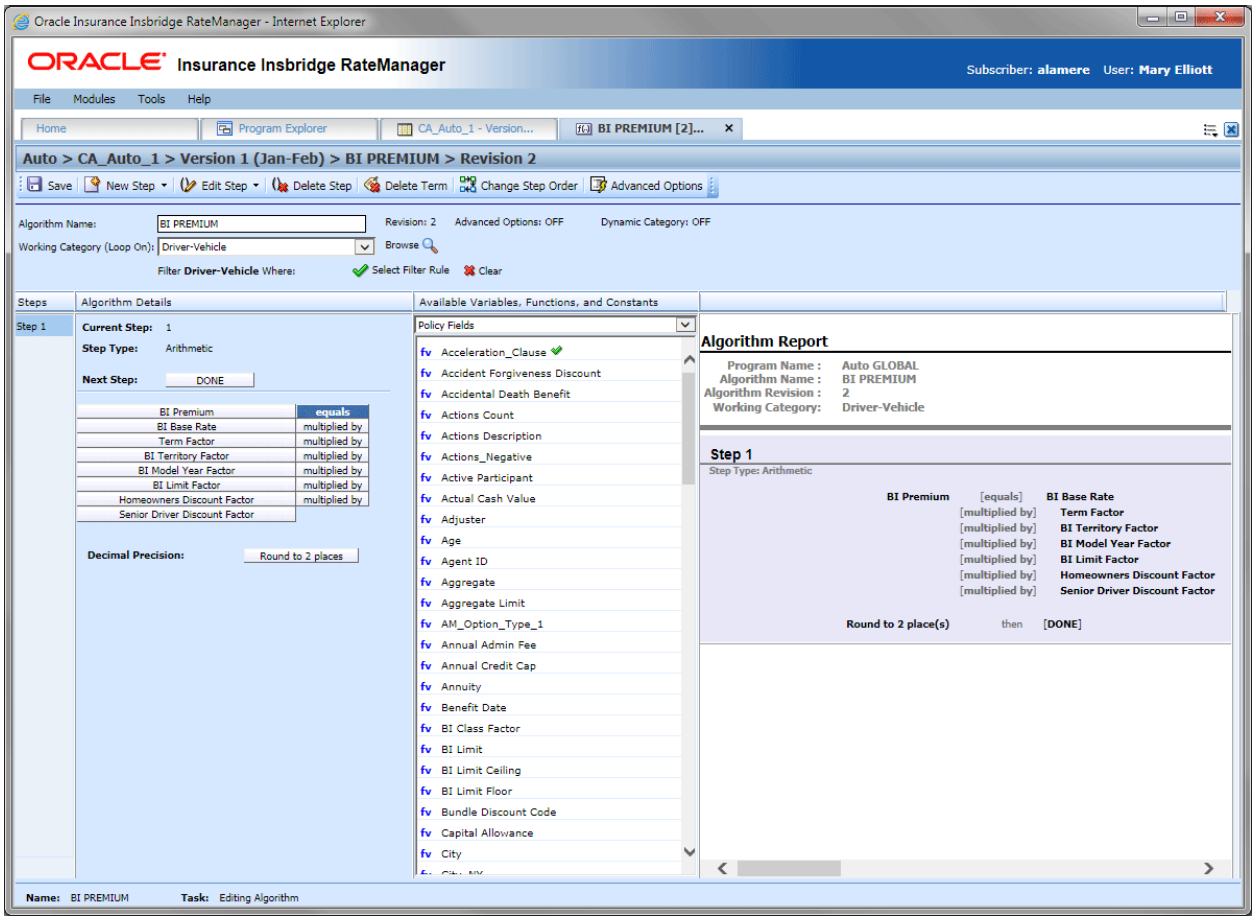
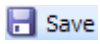
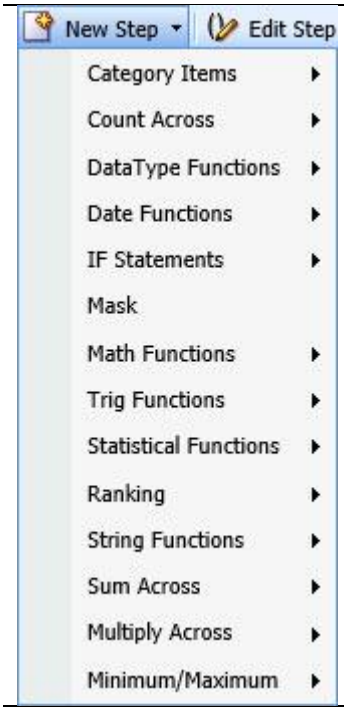
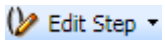
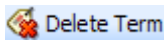
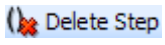
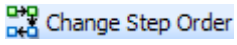
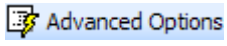


Figure 373 Algorithm Tab

ALGORITHM TAB FUNCTIONALITY

The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform.

	**Save:** Saves the algorithm.
	**New Step:** Creates a new step in the algorithm. For more information, see Working with Steps. **NOTE:** *Underwriting Algorithms have a "Set Underwriting to Fail" step type in the list.* **NOTE:** *If a category instance has been used, a data source query step will be available.*
	**Edit Step:** Copies the selected step or pastes the steps on the clipboard in the selected step order.
	**Delete Term:** Deletes the currently selected term from a step in the algorithm.
	**Delete Step:** Deletes the currently selected step. You cannot delete the last step of an algorithm.
	**Change Step Order:** Allows users to change the order of the steps without using the drop and drag feature.
	**Advanced Options:** Opens up the advanced options screen.

ALGORITHM SCREEN DETAILS

There are four areas on the Algorithm Tab.

- **Algorithm Information:** The algorithm information area contains the name and defines the working category of the algorithm.

- **Algorithm Details:** The details area is where you enter the steps that the algorithm will perform.
- **Available Variables, Functions and Constants:** The list of the available variables, functions and constants that can be used in the Algorithm Details area.
- **Algorithm Report:** A listing of all the steps currently in the algorithm.

Algorithm Tab Algorithm Information

The algorithm information area is where you define the algorithm.

Naming standards can be applied to system elements. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the element and for case enforcement to be applied to any newly created element or any existing elements that are being edited. If a naming standard is in place, the pattern is displayed in the element name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented for all elements.

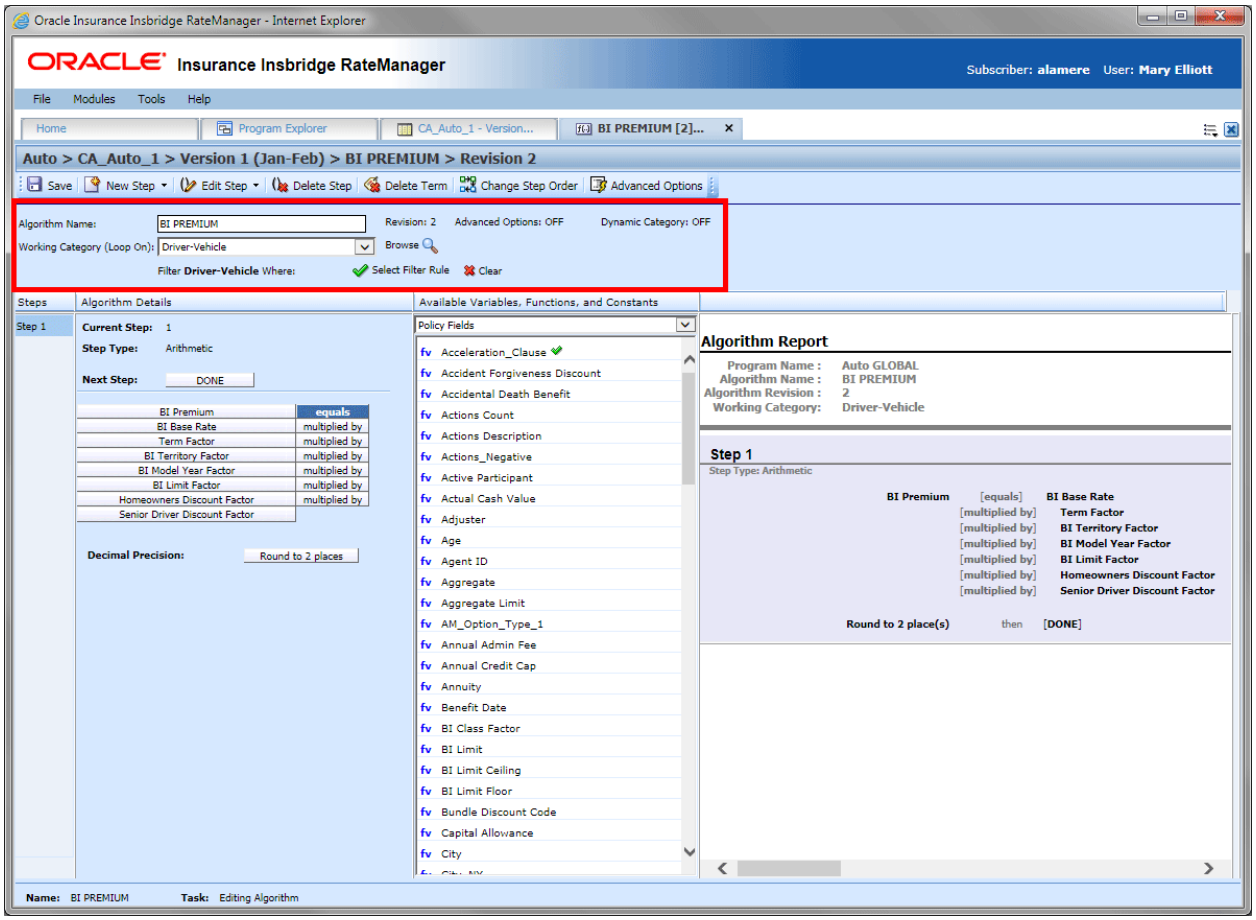


Figure 374 Algorithm Information Field

Screen Details

Algorithm Name: Entry for what you want the algorithm to be called. Alphanumeric characters are permitted, but no special characters (ex: &, *, ", +, @, etc.) are allowed. If a naming standard

is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

Working Category: Sets the category where the algorithm will be run. You can select the working category from the drop down menu or you can browse for a working category by clicking **BROWSE**. This brings up the Browse Category window. From here you can navigate to the Category you want. Once you have selected the category, click OK. The category will auto fill in the working category section.

The Working Category determines the options displayed in the Available Algorithms area. For example, if you select Policy as the Working Category, then the menu choices in the Available Variables area will be Functions, Constants, Step Variables, Policy Fields, Policy Variables and Policy Results.

Advanced Options: Displays an ON if there is an advanced option on the algorithm or OFF if no advanced option is displayed.

Filter Rule: Filter rules are assigned to table variables, calculated variables, or algorithms to determine if the algorithm should be run. Click the green checkmark to add a filter rule. The rule can be removed by clicking the clear option. When a rule has been selected, the rule name is listed.

Dynamic Category: Displays an ON if there is a dynamic category function in the algorithm or OFF if no dynamic category function has been added.

Algorithm Tab Algorithm Details

This area contains the steps that the algorithm will perform.

A new algorithm always begins with an arithmetic step as **Step 1**. If you do not want an arithmetic step as your first step, add the desired step and then delete the arithmetic step. To do so, highlight **Step 1** and click **Delete Step** in the menu bar at the top of the tab.

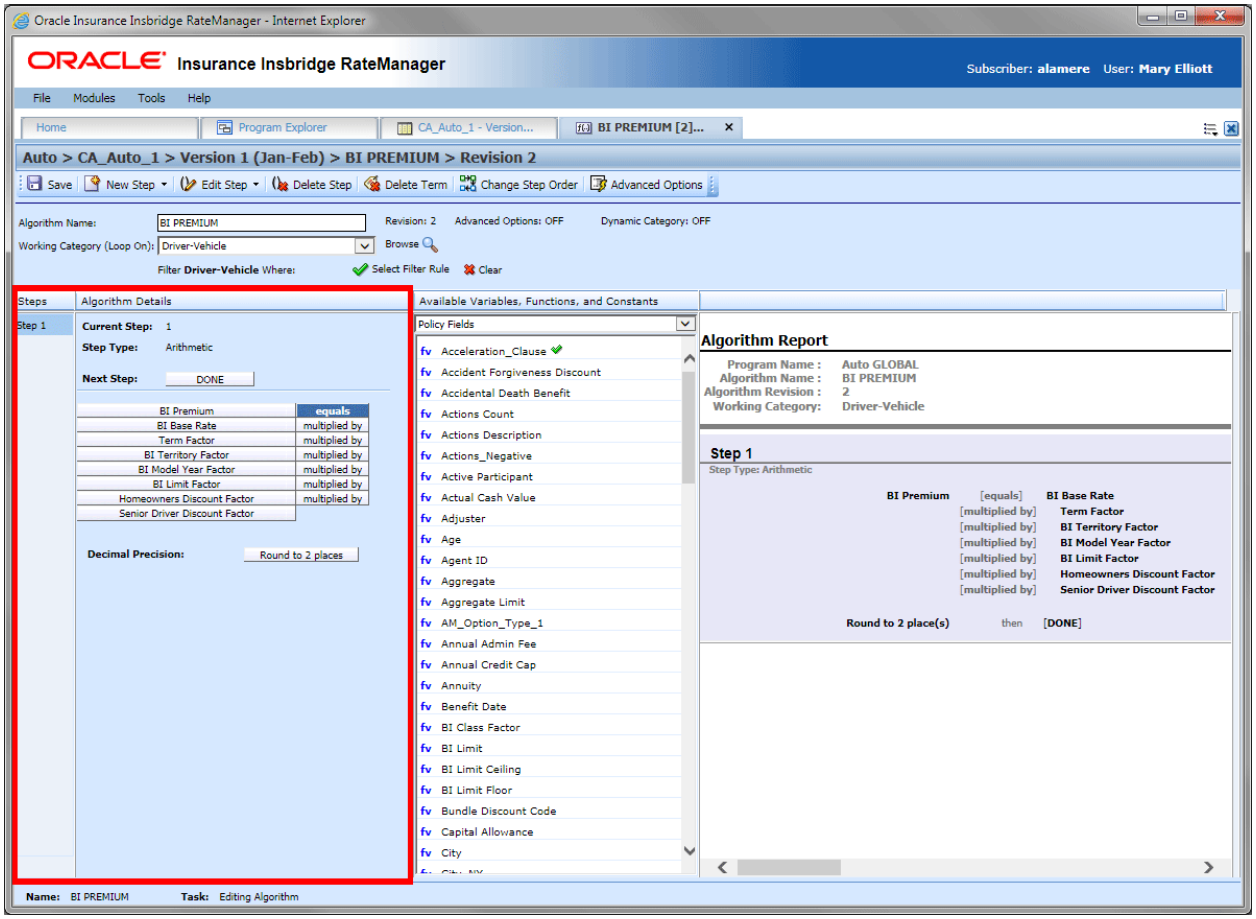


Figure 375 Algorithm Step Details

Screen Details

Steps: Sequential order of steps created with drag-and-drop capabilities for moving a selected step up or down in the sequence.

Algorithm Details: Shows the selected step number, step type selected, next step to follow and fields/variables used for calculation. Rounding is set by clicking on the text box next to **Decimal Precision**. You can select a rounding precision from 0 to 10 digits.

Fields and variables has a right click menu available; View Report, View Where Used and Custom Value.

Algorithm Tab Available Variables, Functions and Constants

The Available Variables, Functions and Constants area contains a listing of available fields, variables, and results, sorted by category; system and domain constants; and message templates that can be used as criteria. For universal algorithms, only universal variables or fields may be used. For global algorithms, only global variables may be used. You cannot use any local variables in the creation of a global or universal algorithm. A globe icon is used to indicate a global algorithm.

For Fields, you can add a new field variable without leaving the Edit Algorithm tab. Scroll to the bottom of the Available Variables, Functions and Constants list and select <New Field Variable>. A screen is displayed that allows you to enter a new field variable.

You also can add new a new result variable without leaving the Edit Algorithm tab.

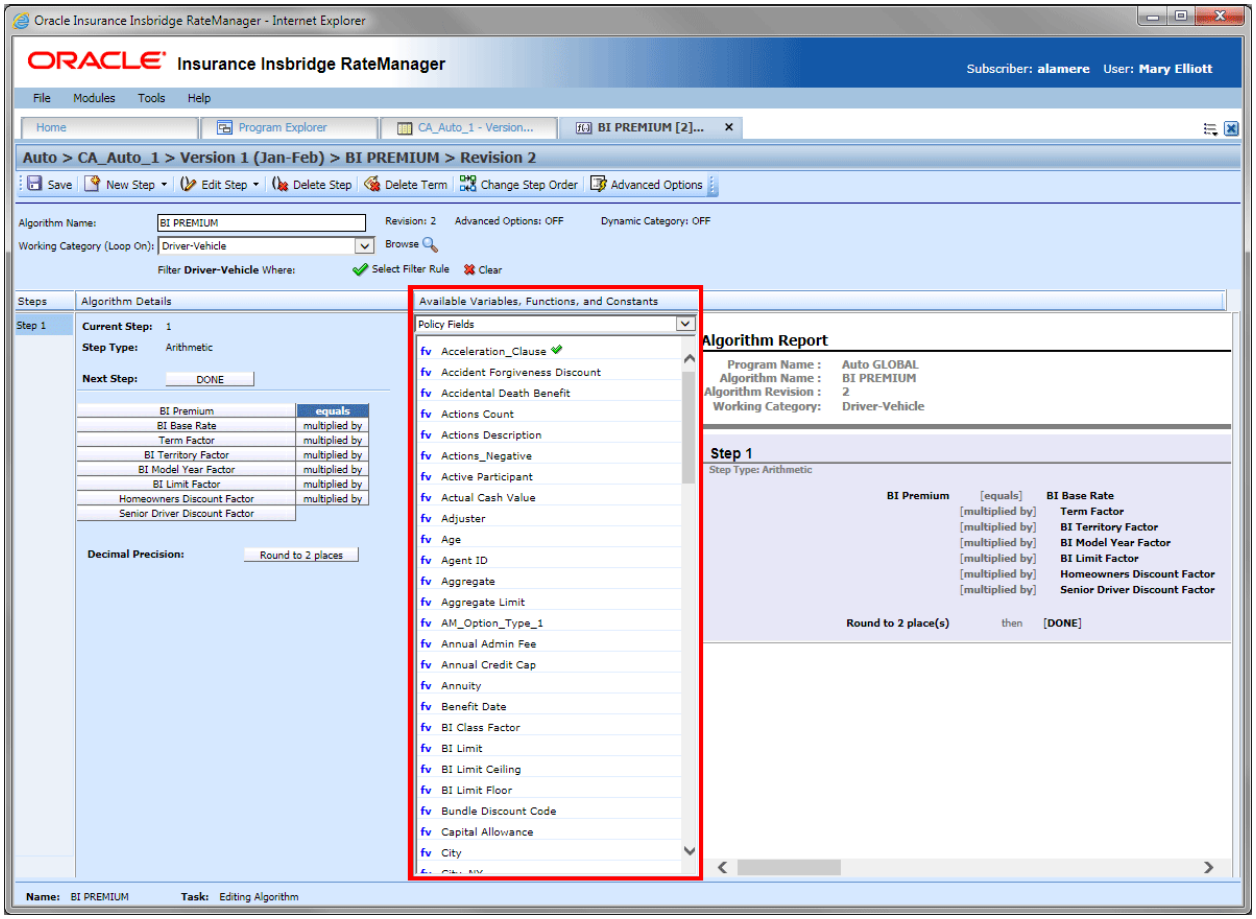


Figure 376 Available Variables, Function and Constants Field

Screen Details

Menu: The menu is the top most field. Click the down arrow to select the category you want. Policy variables are available for all categories. Driver and Vehicle categories are available in the Driver-Vehicle category. Categories are determined in the Algorithm Information area.

Selection: The available variables, functions, constants and step results are listed. Variables are based on the working category you select. Callouts are at the local level only.

Selection	Description
Functions: Global Versioning	Select the program with the global versioning you want to use. For use in local algorithms.
Functions: Library Callouts	Select a SoftLibrary that has been registered in the IBFA. For callouts, step types do not matter. The entire step will consist of the callout. Callouts are program specific and are available in local algorithms.
Functions: Program Callouts	Select the program to be called out to. For callouts, step types do not matter. The entire step will consist of the callout. Callouts are program specific and are available in local algorithms.
Functions: Create Category Instance	Select the category where the dynamic category instance should be created. For use in local algorithms.
Category Instance Mappings	Select the mapping to be used with the dynamic category instance.
Message Templates	Message templates are prefabricated messages with variable placeholders that allow users to create standard messages using variable information that is generated at the time of execution.
System Level Constants	Pre-defined constants, including category instances and system date and time, and system wide user-defined constants. Or you can create a custom value for use in this program.
Domain Constants	User-defined constants found only in a selected product.
Step Variables	Results from the previously entered steps.
Policy Fields	Fields in the Policy category.
Policy Variables	Table and calculated variables in the Policy category.
Policy Results	Result variables in the Policy category.
Policy Data Sources	Data Sources in the Policy category.

Using Functions

Algorithms and Driver Assignments have functions available on the Available Variable, Functions, and Constants menu.

When a function is selected, the function is added to the algorithm or drivers assignment as a step. The function step can be moved, deleted, and edited like any other step. Functions are not available from the Steps drop down menu.

Listing may also include other category fields, variables and results.

Algorithm Tab Algorithm Report

This area contains a detailed list of the steps you have entered. This information is view only. No actions can be performed in this area.

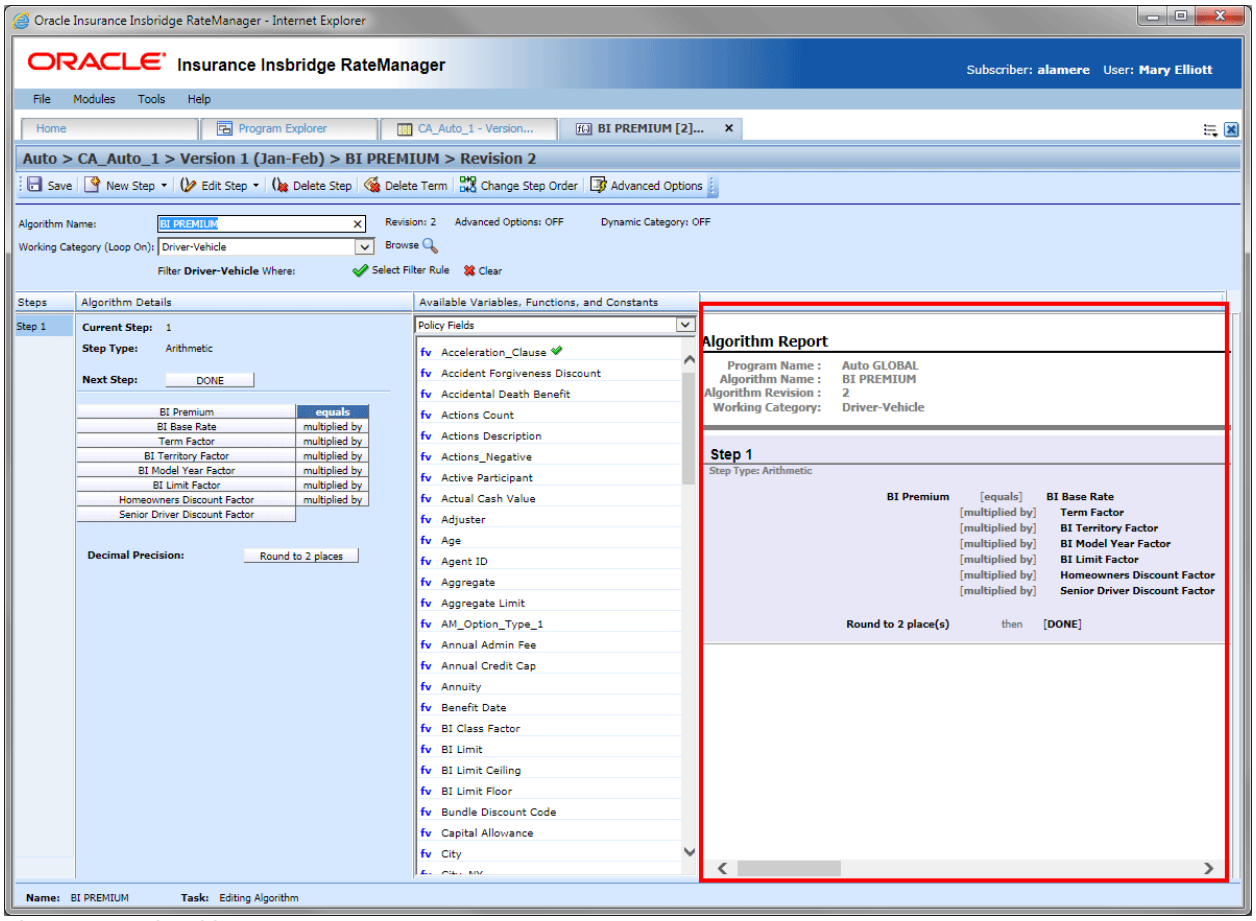


Figure 377 Algorithm Report

CREATING AN ALGORITHM

Algorithms can be created and added to standard programs and template generated programs. Local algorithms are available in the program where you created it and can be copied to other programs within the product. Global algorithms are available across the product. Universal algorithms are available across the project.

When you create a new algorithm, it gets created in all versions of a program. Algorithms can be added to both standard and templates programs. There is no limit to the number of algorithms that can be created for a program.

1. To create a universal algorithm, select the universal icon under the project where you want to create a variable on Program Explorer. To create a global or local algorithm, enter the program details tab from the program where you want to create an algorithm. There are two options; select the global or local algorithm option from Program Details Folder. To open the New Table Variable tab you must select the path for the type of algorithm you want to create.

On the universal tab and the global tab:

New→Algorithm→Normal Rating
New→Algorithm→Underwriting

On the Program Details tab:

New→Globals→Algorithm→Normal Rating
New→Globals→Algorithm→Underwriting
New→Locals→Algorithm→Normal Rating
New→Locals→Algorithm→Underwriting

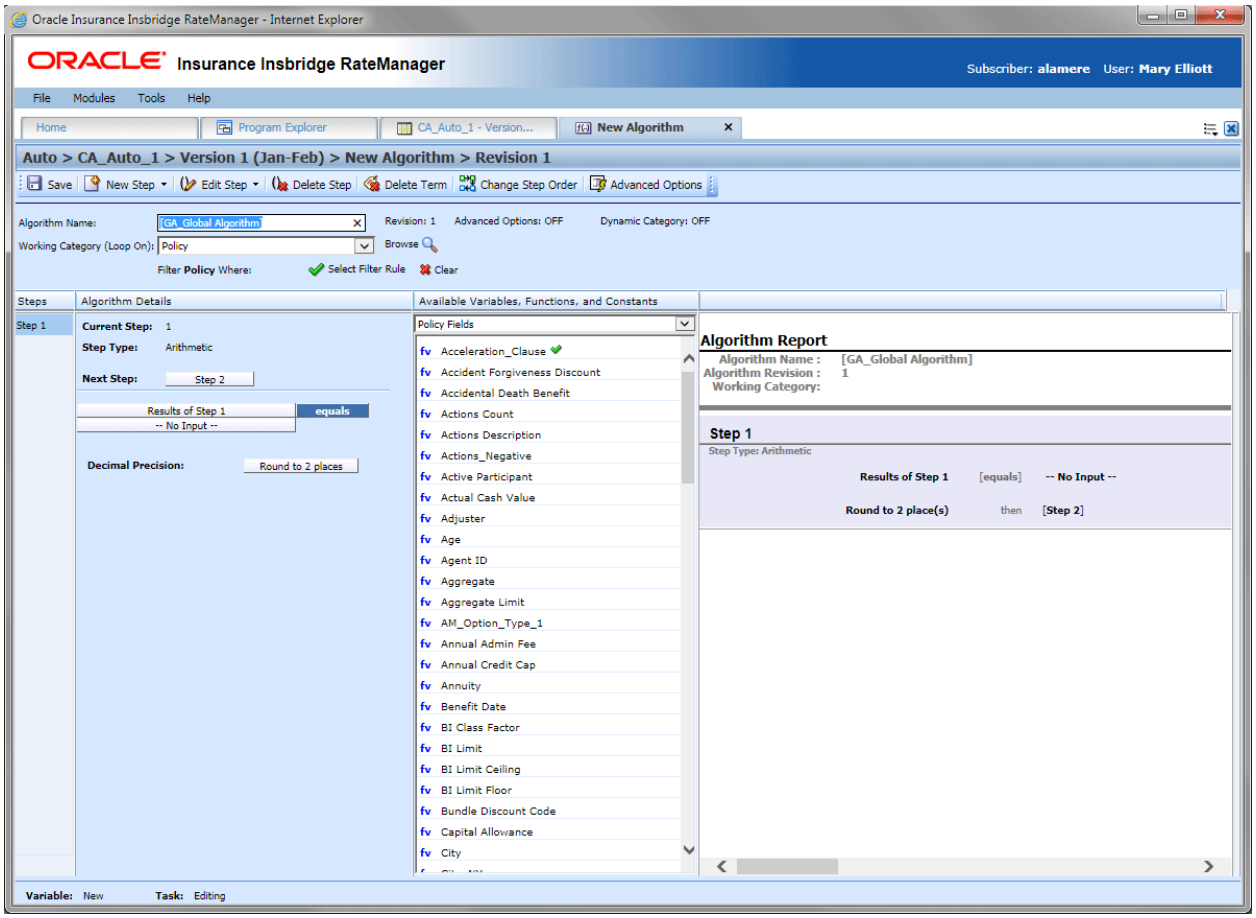


Figure 378 Creating a New Algorithm

2. Enter a **Name** for the algorithm.

3. Choose a **Working Category**. You can select the working category from the drop down menu or you can browse for a working category by clicking **Browse**. This brings up the Browse Category screen.

   From here you can navigate to the Category you want. Once you have selected the category, click OK. The category auto fills in the working category section.

   If this is not the screen you want, click **Cancel** to close this screen and return to the previous screen.

NOTE: *It is a good practice to save an algorithm after defining the name and working category.*

4. The first step is populated in the Algorithm Details area. This must be a mathematical step. If you do not want an arithmetic step as your first step, add the desired step and then delete the arithmetic step. To do so, highlight **Step 1** and click **Delete Step** in the menu bar at the top of the tab.

5. To add a new step to the algorithm, click **New Step** and select the type of step you want to add. The screen refreshes with the new step parameters. Depending upon the step type, you can select the appropriate variables, functions, constants or fields. The values required are listed on the step.

   For example, if the step has an “Enter Variable” box, you must choose the variable type from the menu in the Available Variables, Functions and Constants area. The options are populated underneath. Click the “Enter Variable” box and then double-click the corresponding variable you want. The box refreshes with your selected value. For more information on the types of steps, see Working with Steps.

NOTE: *For callouts, step types will not matter. The entire step will consist of the callout. No other entry can be made. Callouts are program version specific and will be available in local algorithms and driver assignments only.*

6. Check the filter rule option to add a filter rule. See Adding Filter Rules for more information

7. Continue entering steps until complete. When you are finished making your entries, click **Save**.

Adding a Variable, a Result Variable or a Custom Value

A new field variable, result variable, or custom value can be created while entering or editing an algorithm, without leaving the Edit screen.

- To add a field **Variable**: See Adding a New Field from Another Area
- To add a **Custom Value**: See Custom Values
- To add a **Result Variable**: See Adding a Result Variable from Another Area

Using an Advanced Option

The advanced options screen is where you can activate Advanced Algorithm Looping. This option allows an algorithm (and all of its steps) to be processed 1 or more times on a given category instance. Processing will execute until the Do While conditional expression is not met or a Maximum Number of Loops has been executed. The Maximum Number of Loops is required to prevent an infinite loop from occurring. The default number of loops is zero (0).

You can add multiple Do While conditions. The maximum number of loops that the system will allow is 5,000. Be aware that this number may be decreased by your system administrator. You must check the TURN ON ADVANCED LOOPING box and enter a maximum number of loops if you want looping to occur.

NOTE: *Selecting EXIT LOOP as a Next Step will exit the algorithm after the step is finished and prevent any further iterations of the algorithm from executing on any other categories instances at the same level (working category).*

Advanced Algorithm Looping can be turned ON or OFF only on the Algorithm Advanced Options screen. When you turn the algorithm loop off, you will be asked if you want to clear the looping condition. If you answer yes, the screen will clear. Advanced Algorithm Looping is available on normal rating algorithms only.

Algorithm Looping is typically used in conjunction with Program to Program (P2P) or Program to Library (P2L) Callouts. Advanced options are not available on underwriting rules.

To Add an Algorithm Loop to a Rating Algorithm

An Algorithm Loop can be added at any time and on any step in the algorithm. It is not a step type. Looping applies to the entire algorithm.

Advanced Options cannot be added to a new algorithm until you save your algorithm.

1. Create a new algorithm or navigate to the **Edit Algorithm** screen for the program that contains the algorithm where you want loop to occur.

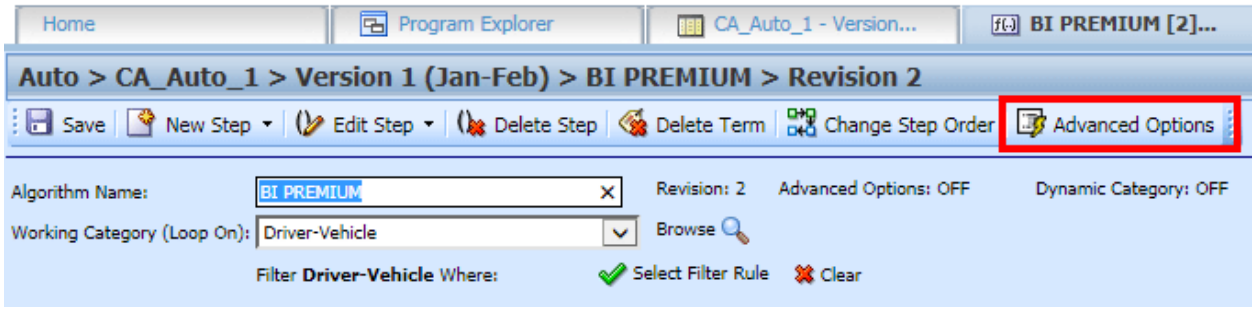


Figure 379 Adding an Advanced Option to a Rating Algorithm

2. Click **Advanced Options**. This opens the **Algorithm Advanced Options** screen.

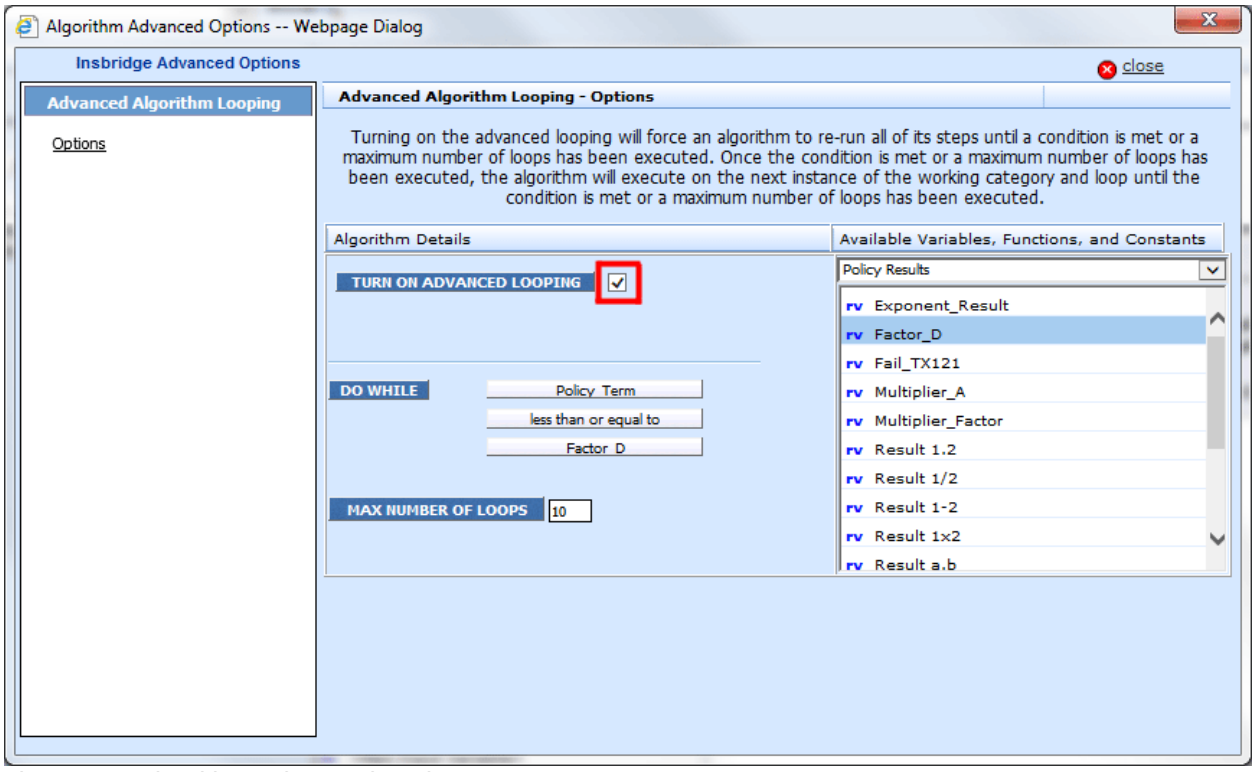


Figure 380 Algorithm Advanced Options

3. Check the **TURN ON ADVANCED LOOPING** box. This will turn on the algorithm looping.
4. The **DO WHILE** section is where you will select the conditions of the loop. The first variable must be a result or field variable. Select the **first variable**.
5. Equal is the default operator. If you need a different **operator**, select it.
6. The second variable can be any variable at the same working category or parent category. The second variable can also be a constant. Select the second **variable**.
7. Enter in the **maximum number of loops** you want to perform. The default is zero (0).
8. Click **Close** to close the screen.

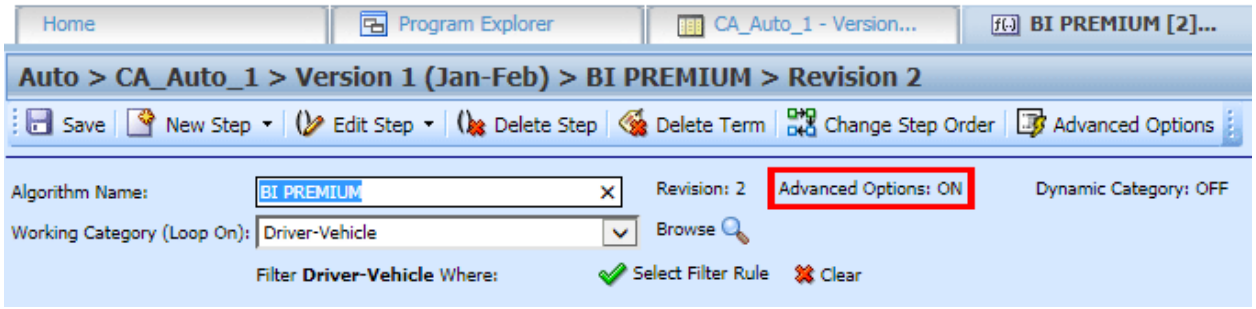


Figure 381 Advanced Options ON

9. On the Algorithm Edit screen, the Advanced Options shows **ON**.
10. Click **Save** to save your work.

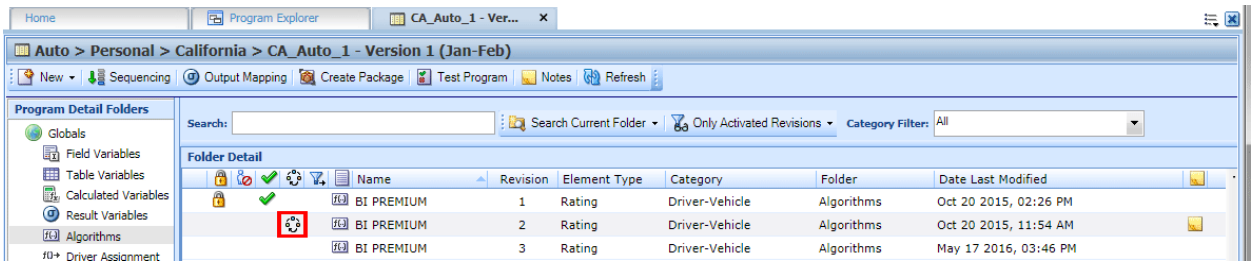


Figure 382 Algorithm With Algorithm Looping Applied

11. On the Program Details tab, the Advance Options Icon is displayed on the algorithm.

Create Category Instance Function

Dynamic category instances provide the ability to control the creation of category instances during program execution. For example; you need to make sure that the XML input contains the prior terms for a span of five years prior to the effective date of the current term. You know some of the prior terms are passed in as a part of the Insbridge XML but, the prior term length may not span the required five years back from the current term. One possible solution is to create a dummy prior term category and set date inputs. These dummy elements would assure that the correct number of prior terms is used. You also would need to make sure that the other miscellaneous inputs on the dummy term are set with default data. This assures that there are records for all 5 prior years but adds a layer of complication to your work.

Instead, you also could use the Create Category Instance Function. The Create Category Instance Function is a function on local algorithms that dynamically creates an instance for the needed category. Using the dynamic category instance function results in the correct number of category instances getting created without creating any dummy elements. You will not have to add extra instances to the input XML to get the desired results.

- The Create Category Instance Function is for use in local algorithms only.
- Categories can be dynamically created for a top level category. For example, any category that is a sibling of Policy or a direct child of the current working category. This helps to track and identify where new instances are created and when they are created.
- The Policy category cannot be dynamically created.
- If the current working category is not Policy, only a child category of the current working category can be selected. For example, if the current working category is a sibling of the Policy level category, like Driver, then only the child categories of Driver will be available for selection.
- Child categories can be one level down only. Second level child categories will not be available for selection.
- The Advanced Option, Do While loop, is automatically presented when creating the first dynamic category instance function/step. Only a single Create Category Instance function/step is allowed if the Advanced Option Do While Loop is on. See Do While Loops for more information.
- If more than one Create Category Instance is in an algorithm, Advanced Options Do While looping cannot be used and the Advanced Option Do While Looping screen is not presented.
- Optional mappings can be added or removed from a Create Category Instance.

- Create Category Instances category selection cannot be edited. If you have selected the wrong category, you must delete the step and re-enter using the category you need. The Create Category step can be moved.
- If a Create Category Instance is used, the Query Data Source step will be available. Please see Data Source Query Step for more information.

Category Mappings

Optional mappings can be created to map fields and results. It is not required to create fields as there may be category instances created dynamically that are not used in rating or underwriting calculations, but only used in output mapping results. In this case, you can create category fields and/or results by using a mapping. Category mappings can be created prior to the Create Category Instance being used.

Category mappings are created and managed on the same screen as Callout Mappings. Please see Create Category Instance Mappings for more details.

Creating a Category Instance

This function is only available on local algorithms with more than the Policy category. The Create Category Instance function is its own step.

1. In the local algorithm, select Functions – Create Category instance from Available Variables, Functions, and Constants.

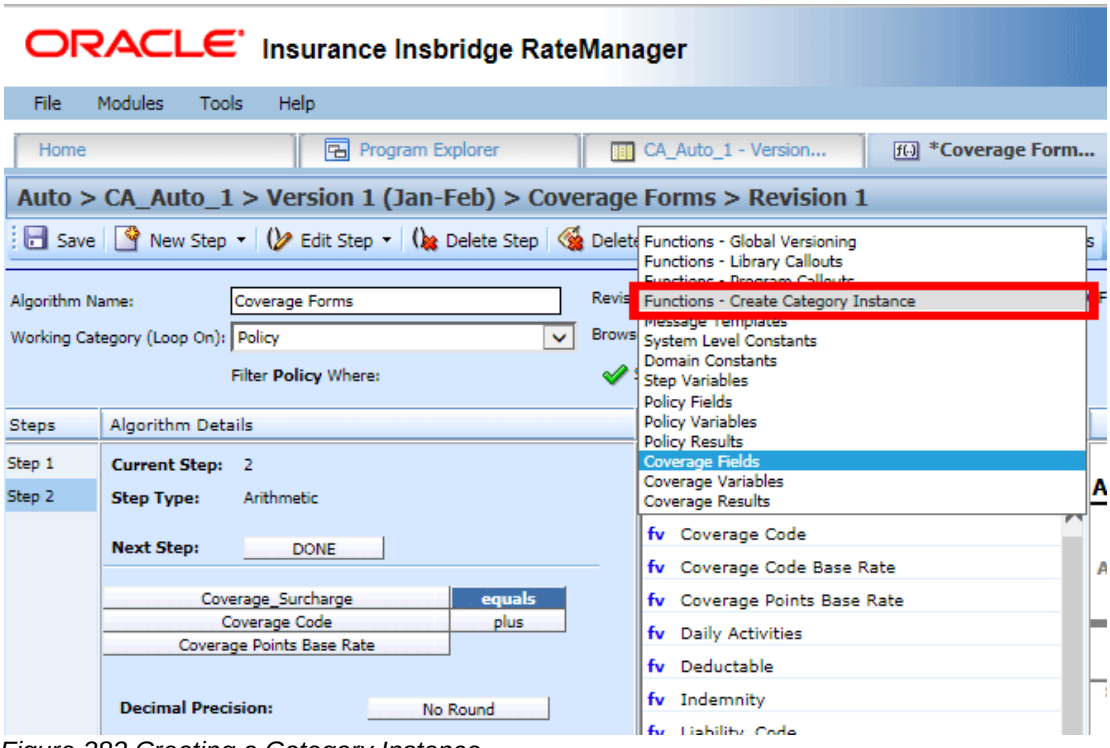


Figure 383 Creating a Category Instance

2. Click the <Select Category> option. The available categories are presented in a separate window.

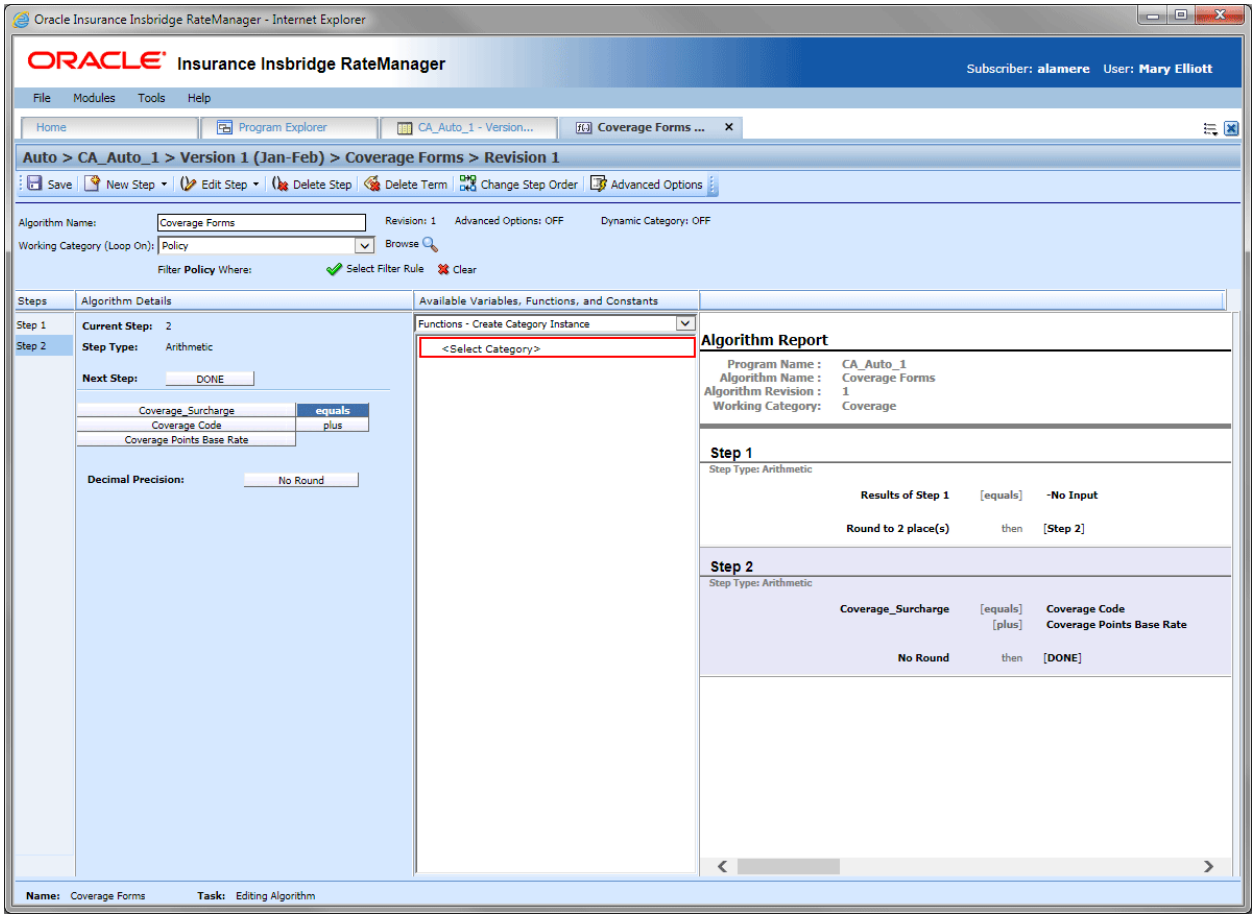


Figure 384 Selecting a Category for Dynamic Category Instance

3. Select the category where you want to create a dynamic category instance.

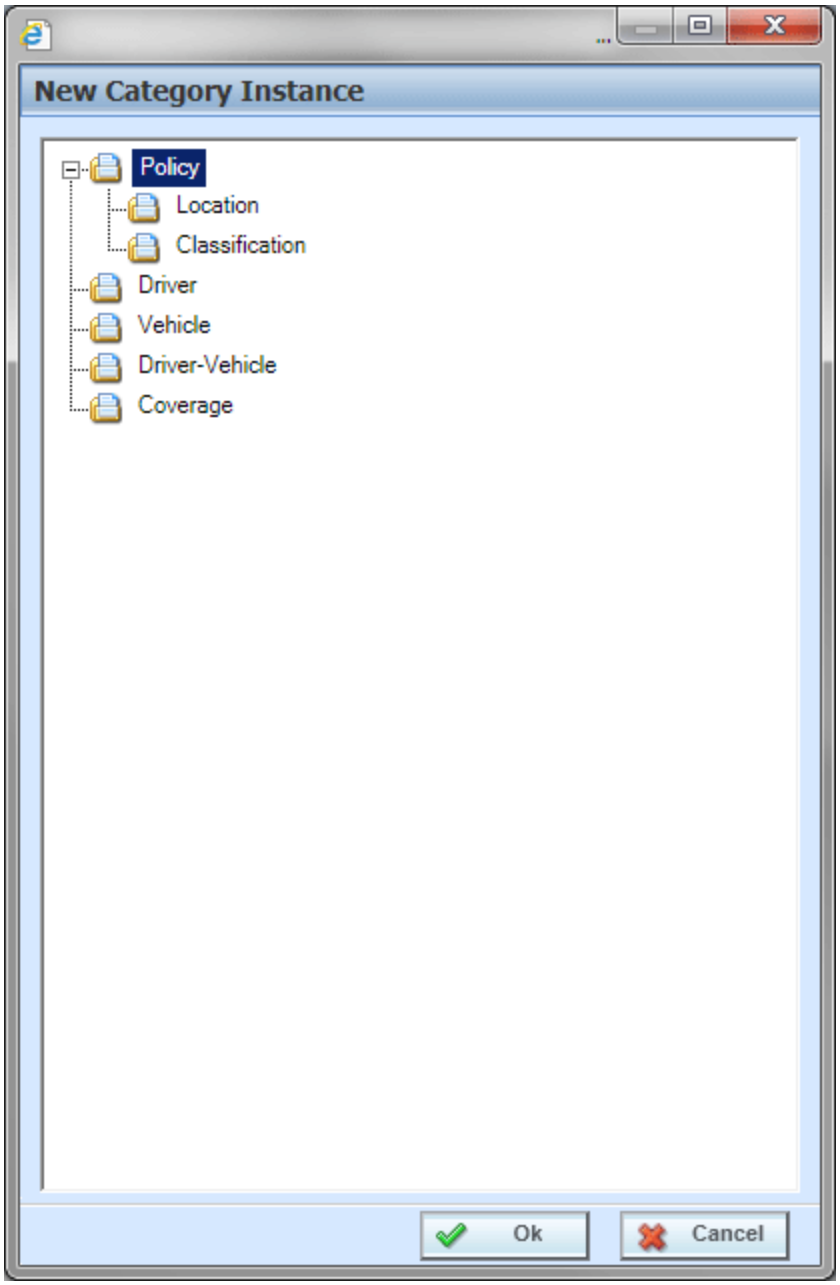


Figure 385 Selecting the Category for Dynamic Category Instance

4. If this is the first Create Category Instance, then the Advanced Options Do While Looping screen is displayed. The Advanced Options Do While Looping screen is not displayed on subsequent Create Category Instances.

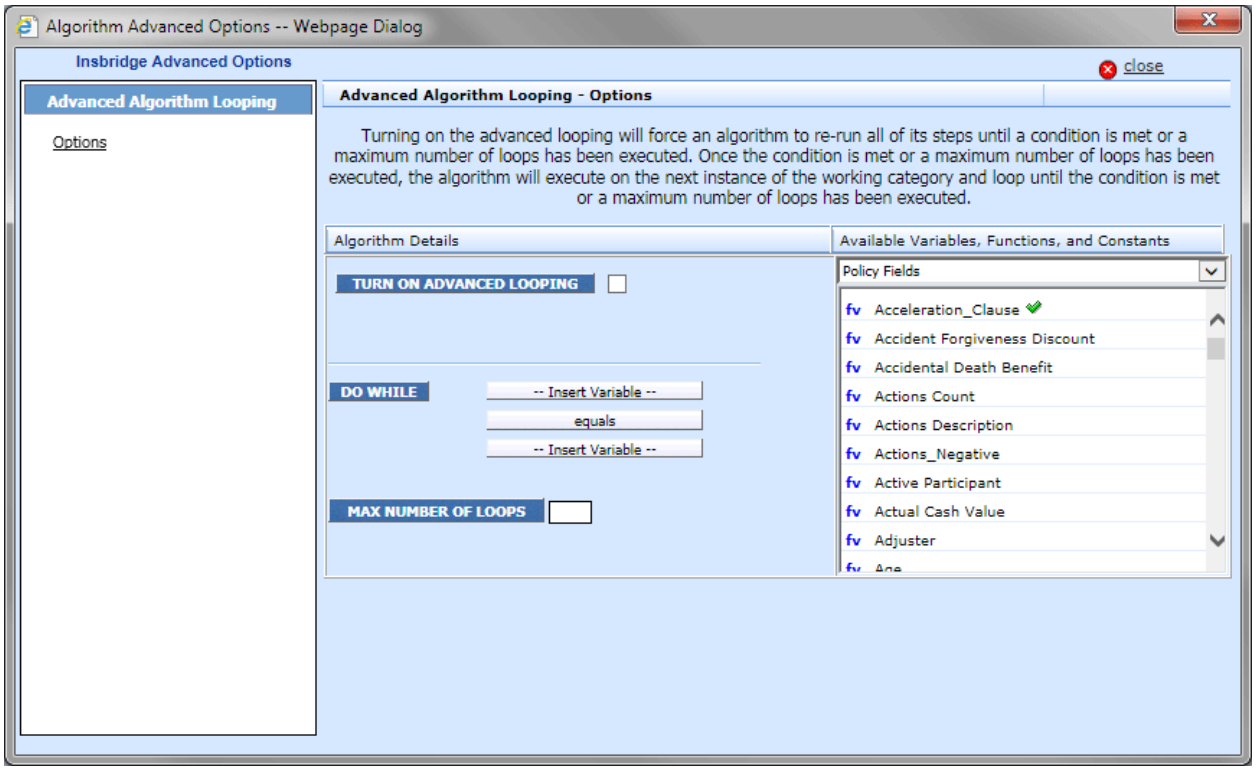


Figure 386 Advanced Options Do While Looping

5. If you want to add Do While Looping, check the Turn On Advanced Looping box and enter the values you need. Only one Create Category Instance function/step can be used if Do While Looping is on.

6. If you do not want to add Do While Looping, close the screen.

7. A new step is created.

8. If an optional category mapping is to be used, click the Use Field Mapping field. Select Category Instance Mappings. Choose the mapping you need. If no mappings are listed, you can save your work and create a mapping.

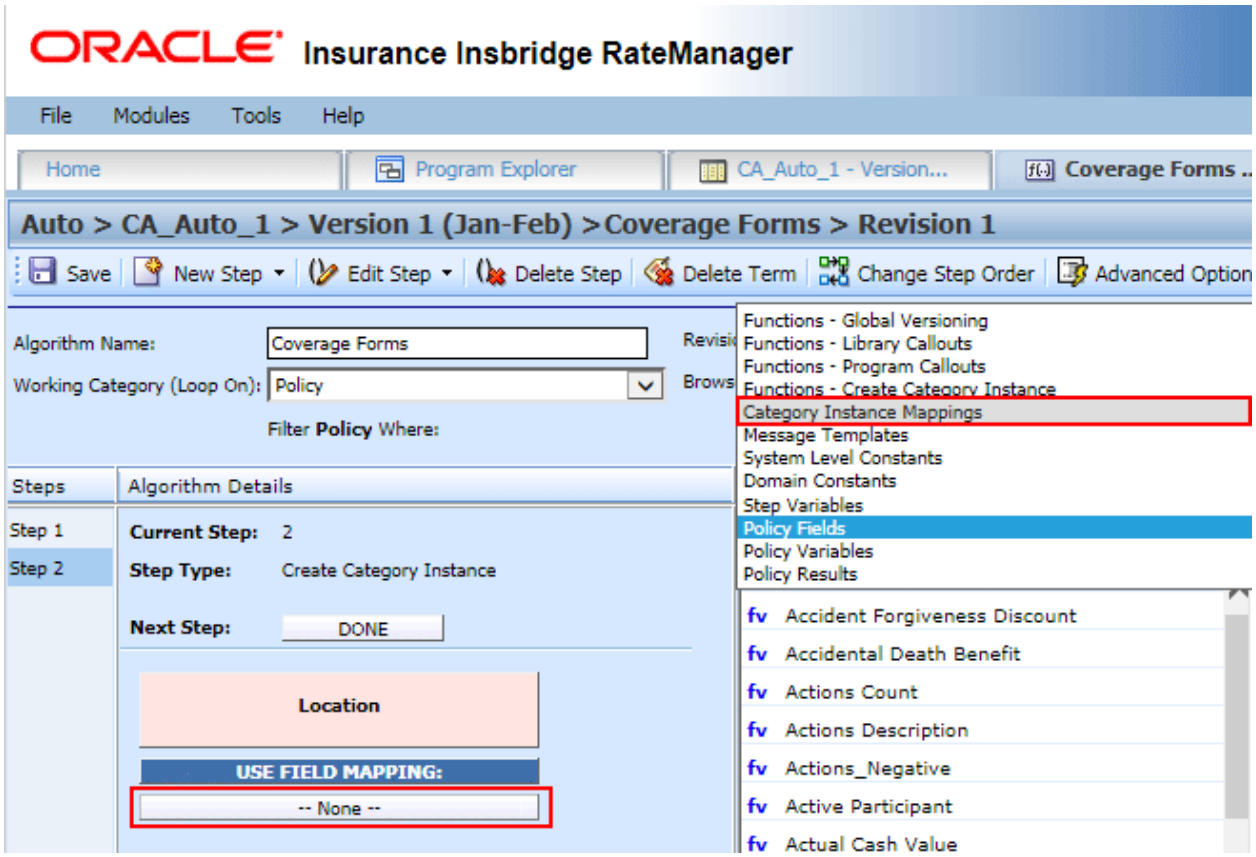


Figure 387 Adding a Create Categories Instance Mapping

9. Save your work. The Dynamic Category option in the Information area will show ON and the Dynamic Category icon will be in front of the algorithm on the Program Details tab.

Editing a Create Category Instance

Only mappings can be edited on a Create Category Instance. If you have selected the wrong category, you must delete the step and re-enter. The new step can be moved.

A Create Category Instance can be copied using the Edit Step button in the top bar menu. The Create Category Instance cannot be copied if a Do While - Advanced Option is being used.

Deleting a Create Category Instance

Create Category Instance steps can be deleted at any time.

- If a Query Data Source step is included in the algorithm, that step must be deleted as well.
- Deleting the Create Category Instance does not remove the Advanced Option Do While looping. If you do not need Do While looping, you must click the Advanced Option button in the menu and uncheck the Turn on Advanced Options DO While Loop.

Dynamic Category Instance Indicator

A system level constant that identifies dynamically created category instances is available in the System level Constants listing. Count Across step types and Sum Across step types can use the Dynamic Category Instance indicator to count the number of instances created or total the number of instances created.

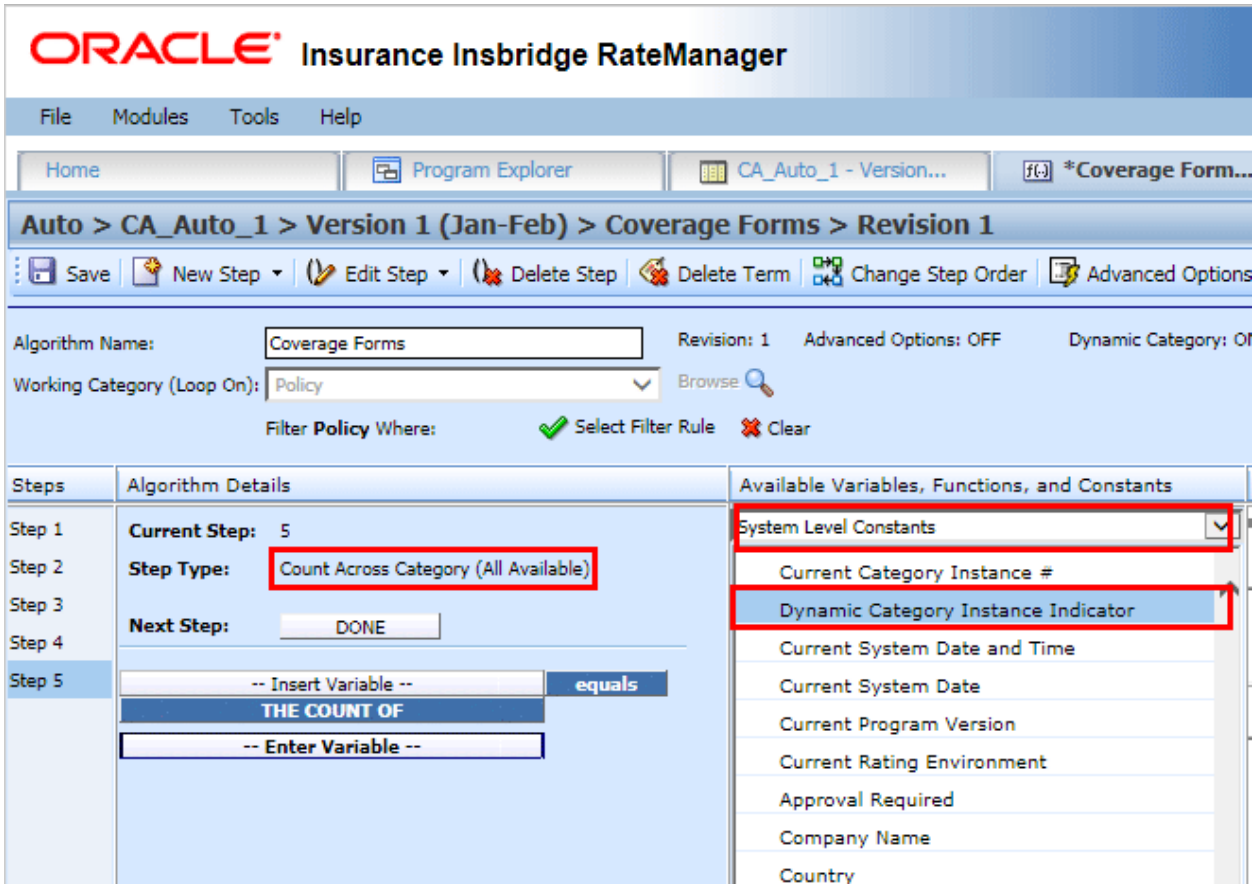


Figure 388 Using the Dynamic Category Instance Indicator

This option is available in the System Level Constants option.

Dynamic Categories on Program Details Tab

An icon is added to local algorithms on the Program Details tab to indicate that the algorithm contains a Dynamic Category step.

Creating a Category Mapping

Mappings are not required when creating category instances but can be added if needed. Category instance mappings map fields, results and constants.

It is not required to create fields as there may be category instances created dynamically that are not used in rating or underwriting calculations, but only used in output mapping results. In this case, you can create category fields and/or results by using a mapping. Category mappings can be created prior to the Create Category Instance being used.

The process of creating a category instance follows the same process as callout mappings.

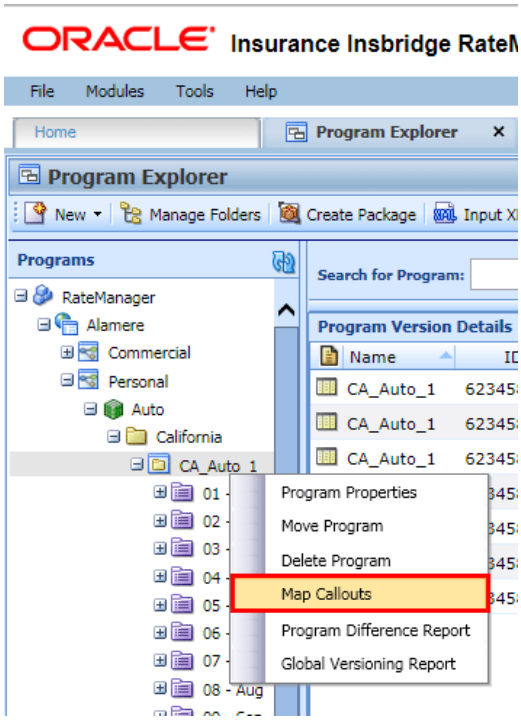


Figure 389 Selecting Map Callouts

1. Select the program where you want to create a category instance mapping.
2. Right click and select Map Callouts. A separate screen opens.

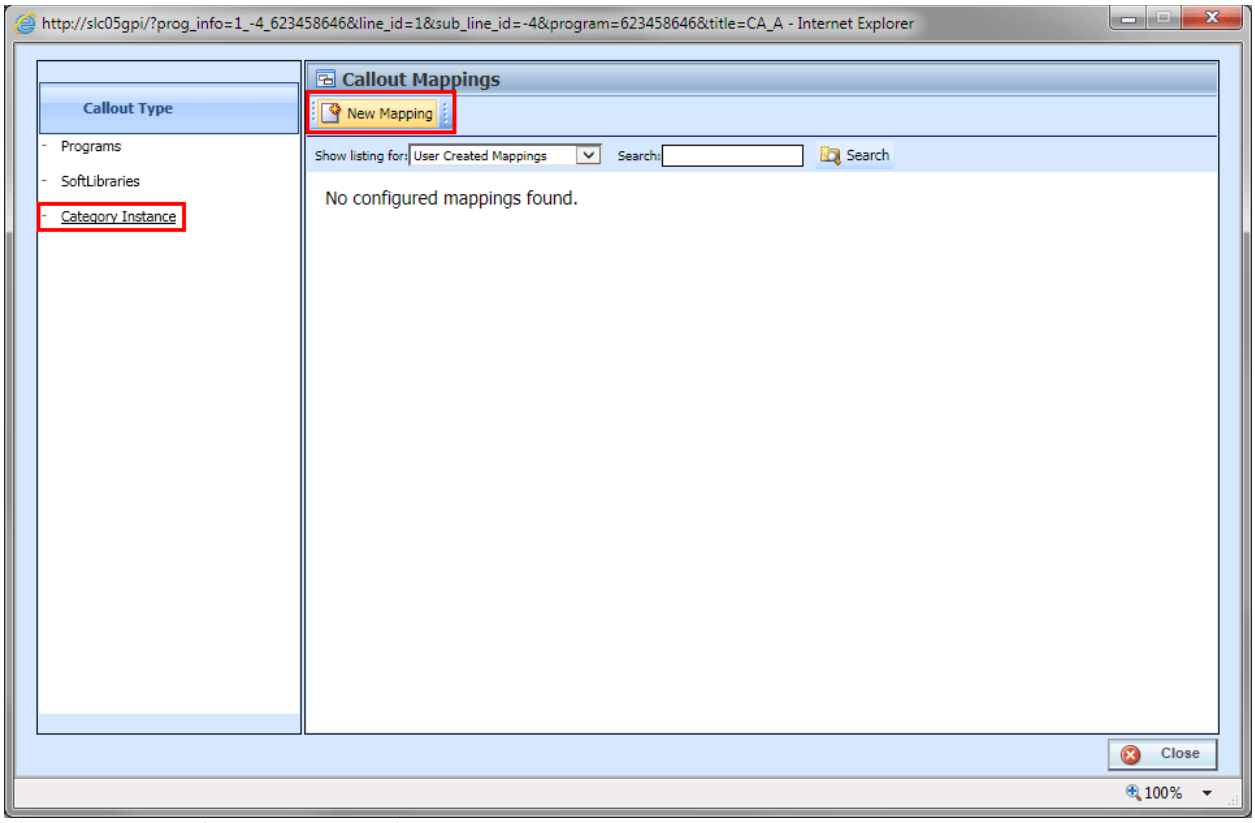


Figure 390 Creating a New Mapping

3. Select Category instance in the left side menu. Any mappings are listed. You can edit or copy a current mapping.

4. To create a new mapping, click the New Mapping button in the top menu.

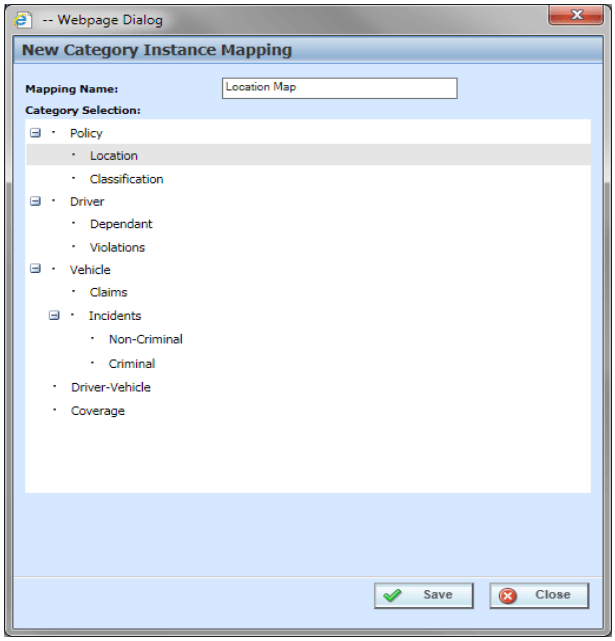


Figure 391 Selecting the Category

5. Enter a mapping name.

6. Select the target category. Save your work.

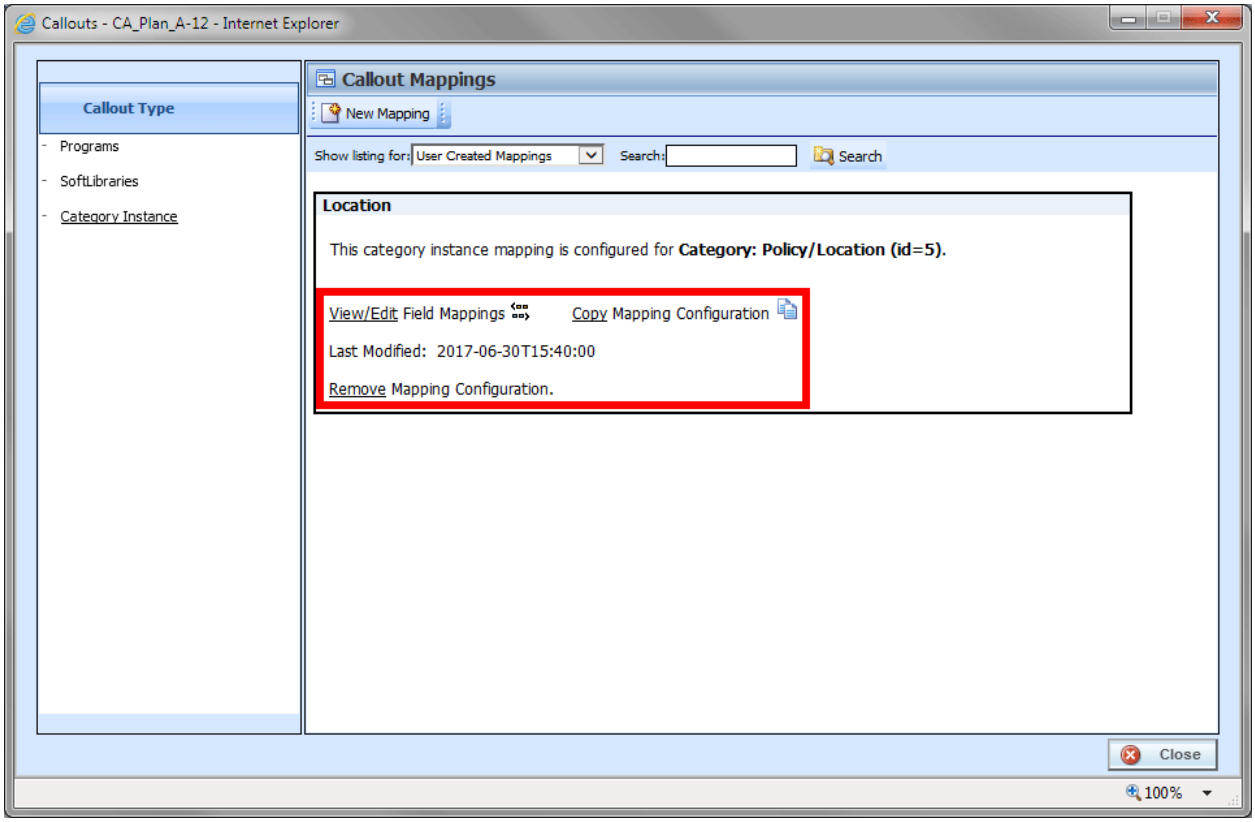


Figure 392 View/Edit Mappings

7. The mapping is listed on the main page. Select the View/Edit link to add mappings.

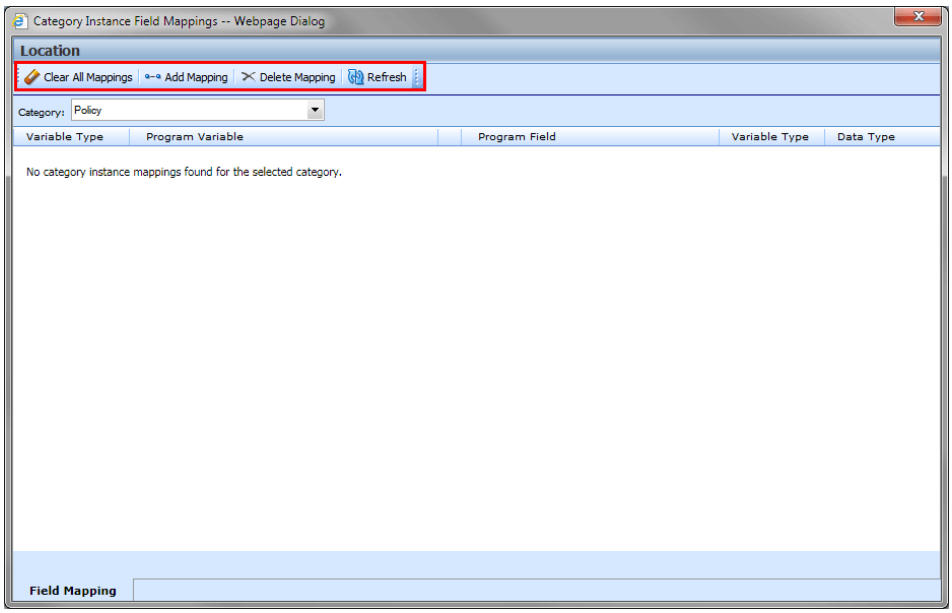


Figure 393 Selecting a Add Mappings

8. For new mappings, click Add Mapping.

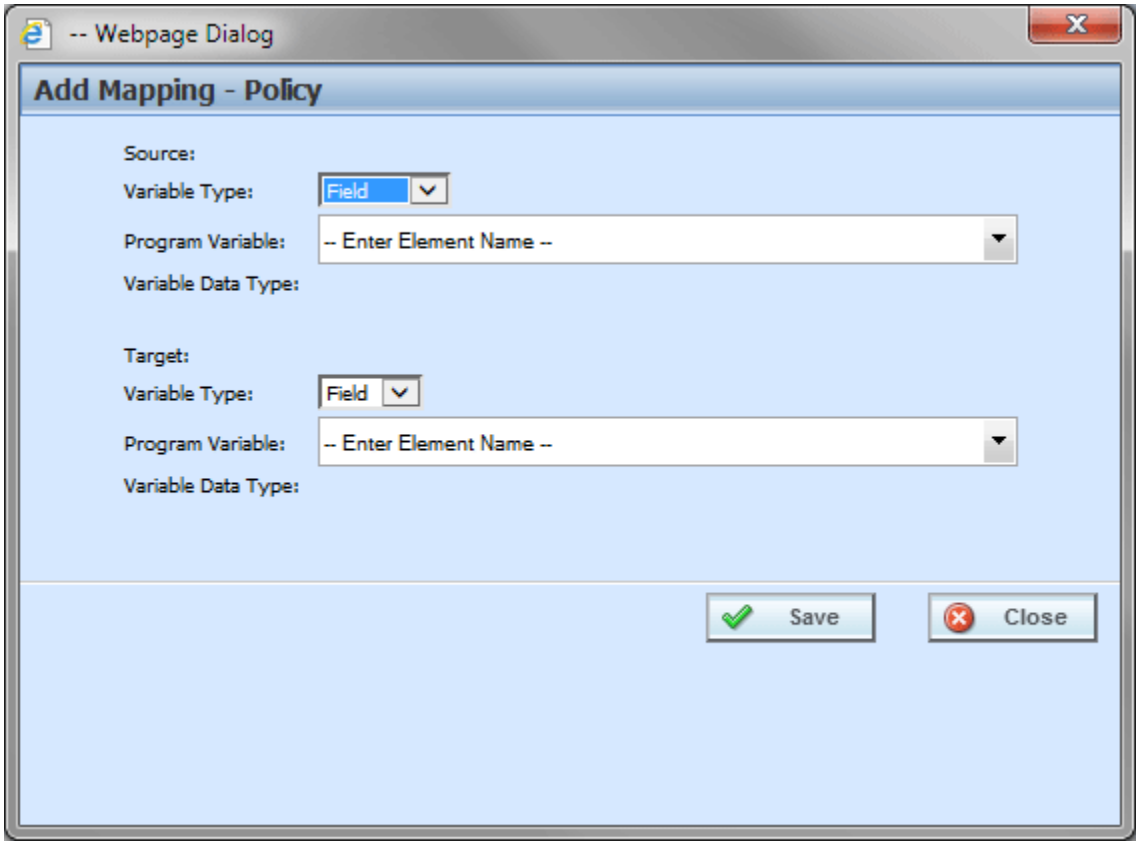


Figure 394 Adding a Mapping

9. Add the source to target mapping.

   Source – current category
   - Mappings can be done on Fields, Constants, and Result Variables
   - P2P returned result values are allowed
   - Data types must match

   Target – new category instance
   - Mappings can be done on Fields and Result Variables
   - Data types must match

10. Save your work.

11. To use a mapping in a create category instance, select Category Instance Mapping from the Available Variables, Functions and Constants. Select the mapping you need and assign to the Use Field Mapping entry.

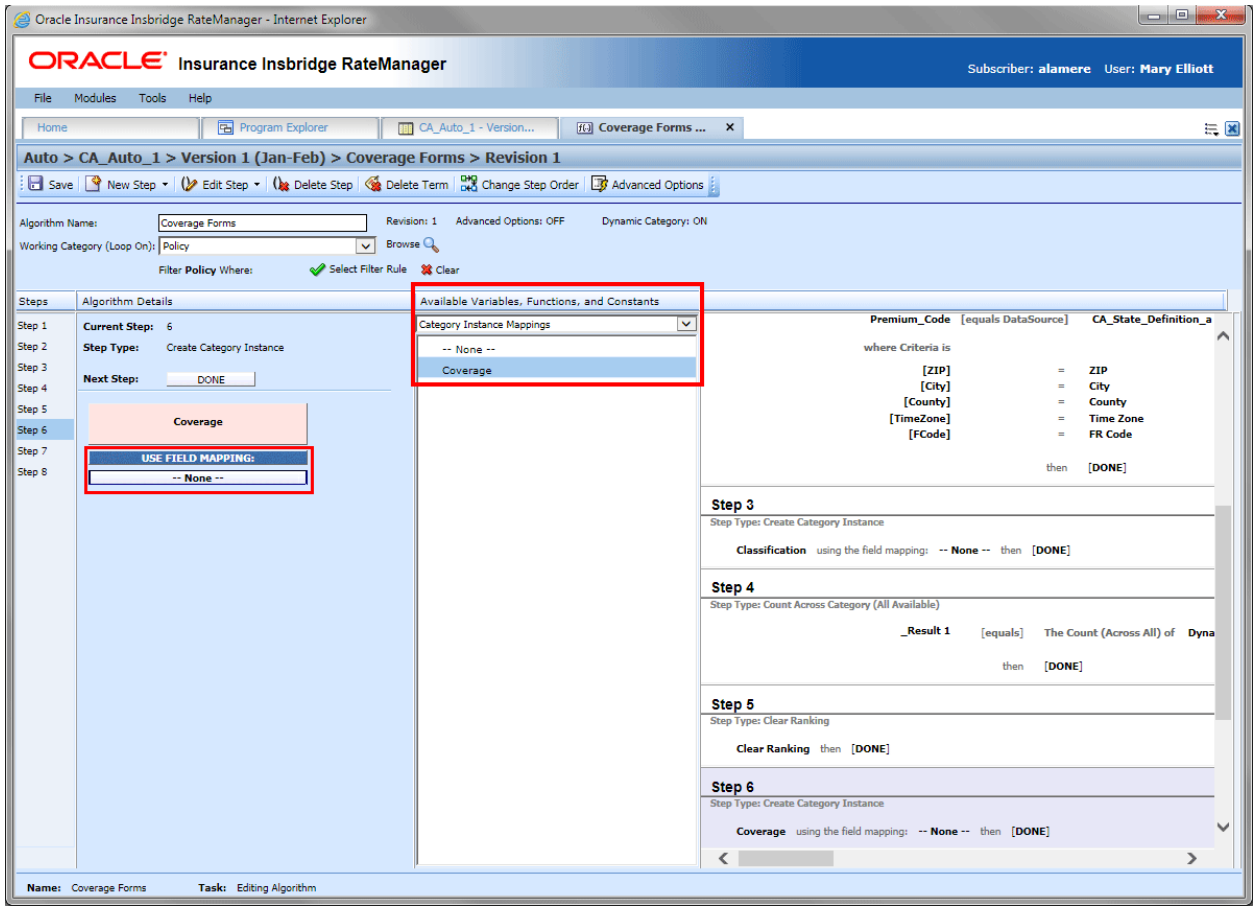


Figure 395 Adding Mappings to Create Category Instance

Mappings are the only editable item on the function/step. If any changes are made to the mapping, the mapping needs to be deleted and re-assigned on the Create Category Instance.

Message Templates in Use

Message templates are user created messages consisting of fixed values and system values that are defined at the system level and available to all projects.

Message templates can be used as a step in any calculated variable, algorithm or driver assignment. Message templates are brought in with text and placeholders where users can assign a variable, function, result or step or constant to be used at execution.

1. Select the calculated variable, algorithm or driver assignment where you want to use a message template.
2. Select Message Templates from the Available Variables, Functions, and Constants.
3. Available message templates are listed.
4. Select the message template you want.
5. The message template is listed as a step.

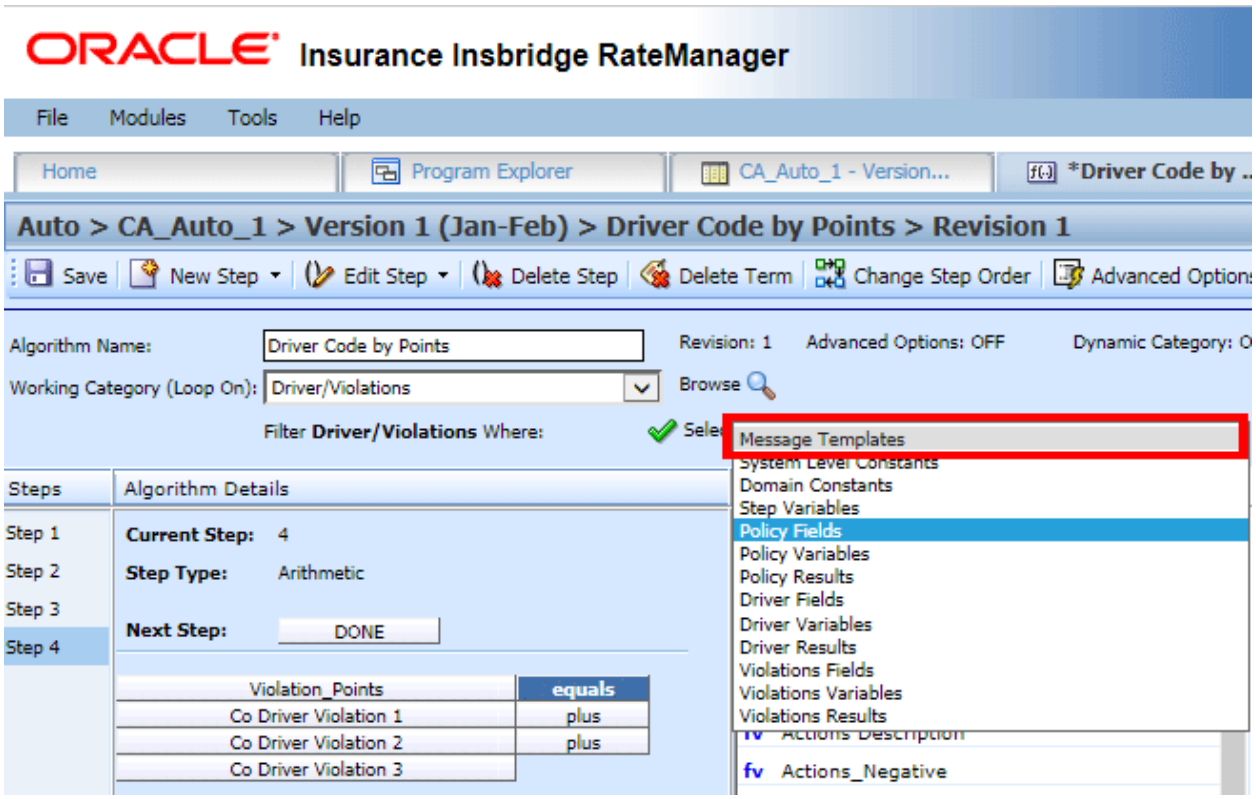


Figure 396 Message Templates

6. You must match up the placeholders [1] with available variables, function, results of step or constants. Select the placeholder and then select the appropriate variables, functions, constants or fields that you want to use.
7. Continue until all placeholders have a value assigned.
8. When you are finished, click Save to save your entry.

EDITING AN ALGORITHM

The Edit Algorithm tab is where you can change algorithm information such as name, working category and steps. Algorithms that are locked cannot be edited. Template generated algorithms that are using revision 1 are not open for editing. If you want to edit a template generated algorithm, you must create a new revision. The new revision will be open for editing except for the name. The name cannot be changed. Global algorithms can be edited from the globals tab or from the program details tab.

NOTE: *If a change is made to a global algorithm, the change affects all programs that use the global algorithm. If a change only needs to be made to one particular program, create a local algorithm in the program and have the program reference the local algorithm instead of the global.*

To Edit an Algorithm

1. On the Program Details tab, select the **Algorithm** you want to edit.
2. Double click or right click and select **Edit**. The Edit Algorithm tab opens with the details of the algorithm.
3. Make any changes.

   To change the step order, you have two options:

   - **Option 1:** Select a step in the Steps column, drag it to a new location in the list and drop it.
   - **Option 2:** Select the step you want to move and click Change Step Order. A separate screen is displayed.
     - The current step position is displayed at the top of the screen. Select where you want the step to be placed.
     - Click OK. You will be returned to the Edit Algorithm tab. Your step should be placed where you chose.

   To delete a step, select the step in the Steps column and then click Delete Step. You cannot delete the last step from an algorithm.
4. Click **Save** to save your work.

WARNING: Category Updates to Revisions

Category updates should not be done to a revision. All revisions of a variable should be at the same category level. Changing the category for only some of the revisions of a variable is not supported.

ALGORITHM COPY OPTIONS

Algorithms may be copied in any program.

- **Local algorithms** can be copied to the same program or to another program at the local level within the same product.
- **Global algorithms** can be copied:
  - At the global level and are available to every program in the product
  - To the local program level
  - To another program within the same product at the local level
- **Universal algorithms** can be copied:
  - At the universal level and are available to every program in the project
  - To the local level of any program within the project

The copy options depend on whether the algorithm to be copied is at the universal level, global level or the local level.

When copying an algorithm, the structure of the algorithm (including the working category and steps) is copied. Only the active revision is copied. If you have an algorithm with 10 revisions, and you copy the algorithm, only the active revision is copied into the new algorithm. The other 9 revisions are not copied.

Universal Copies

When copying a universal algorithm to a local level:

- Universal copies to a local level are project wide. Information has been added to help you select the Destination Program where you want to copy the element to. The Destination Program name is listed first, the folder name is in parenthesis and the product name is last, also in parenthesis.
- Duplicate naming is not allowed. The New Name must be different from the Old Name.
- Product level naming standard are not enforced. If the product has a naming standard, copies made from the universal level are not required to have the naming standard applied when copying. If the variable is edited any time after the initial copy, the naming standard is enforced.
- Universal permissions are required to make copies at the universal level.

Global Copies

When copying a global variable to a local level:

- Global copies are done at the product level. The Destination Program name is listed first and the folder name is in parenthesis.
- The New Name must be different from the Old Name.

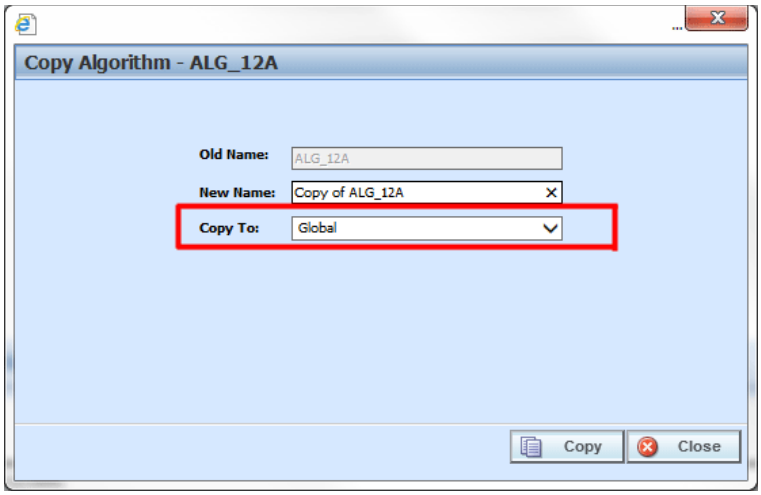


Figure 397 Copying an Algorithm at the Global Level

When a global algorithm is copied to a local level or when a local algorithm is copied, there are two copy options:

- **Existing Program:** Copy the algorithm into the existing program under another name.
- **Other Program:** Copy the algorithm into another program within the same product.

Local Copies

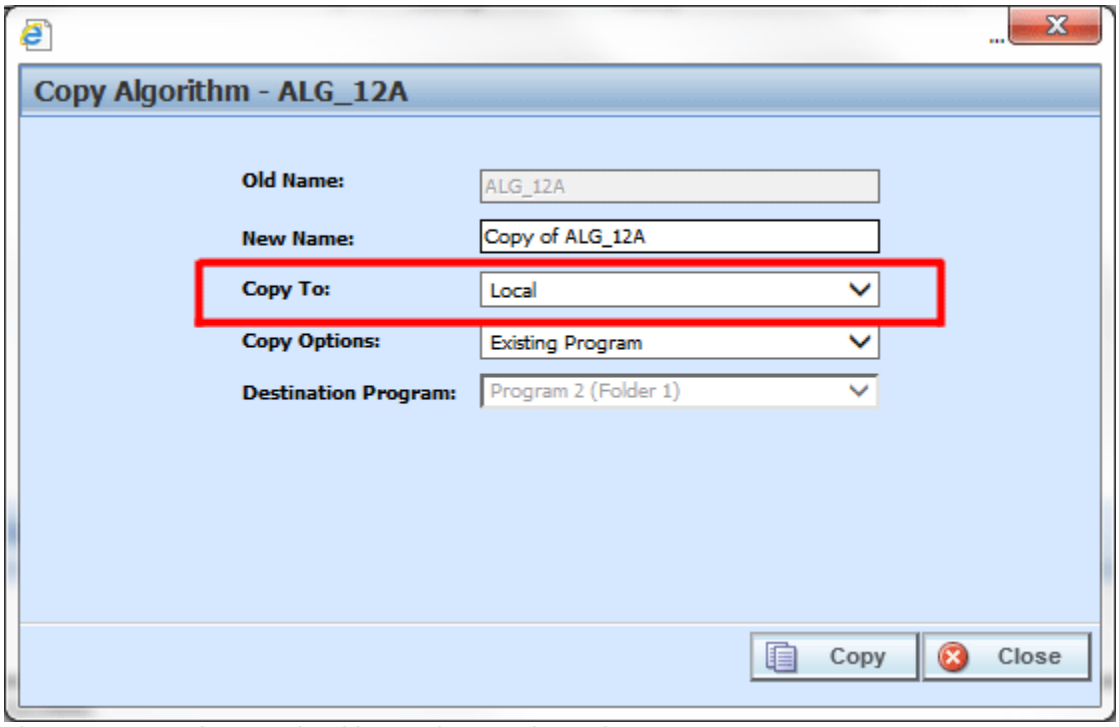


Figure 398 Copying an Algorithm at the Local Level

Local copies can be done to the local level of any program in the product. When copying a variable to a local level in another program:

- Local copies are done at the product level. The Destination Program name is listed first and the folder is in parenthesis.
- Local to local copies check for dependencies when the copy is made:
  - When performing a local to local copy, dependencies such as the usage of other local elements are checked. If there are local elements used in the element being copied that are not in the Destination Program, then those elements are copied also.
  - For example, the Option AB Fee algorithm, a local variable, uses variables Option A and Option B, also local variables. When Option AB Fee is copied from program Maine to program Kansas, the algorithm is copied and so is the variables that are being used.
- Local to local copies do not require a name change:
  - When copying to the local level of a different program, the variable name cannot be changed. The New Name is grayed out. This creates an identical variable in the other program and reduces the need for editing. If a change is made to the original variable and then the variable is copied for a second time, the copy is created as a new revision in the Destination Program.
  - When copying at the local level in the same program, the name must be changed.
- Copies made in the current program must have a new name.

COPYING AN ALGORITHM

Global algorithms can be copied from the globals tab or from the program details tab. Unlike a program level algorithm, you will not be able to select copy options.

1. Select the algorithm you would like to copy and right click. Select **Copy** from the menu.

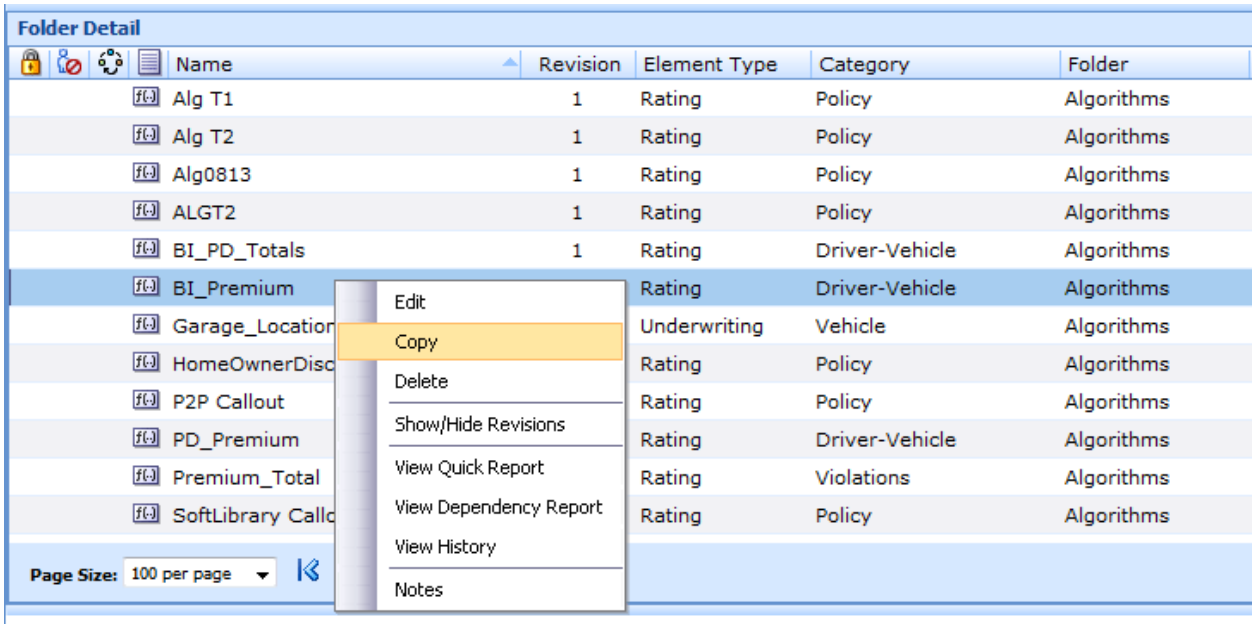


Figure 399 Copy Algorithm Menu

2. A separate window is displayed.

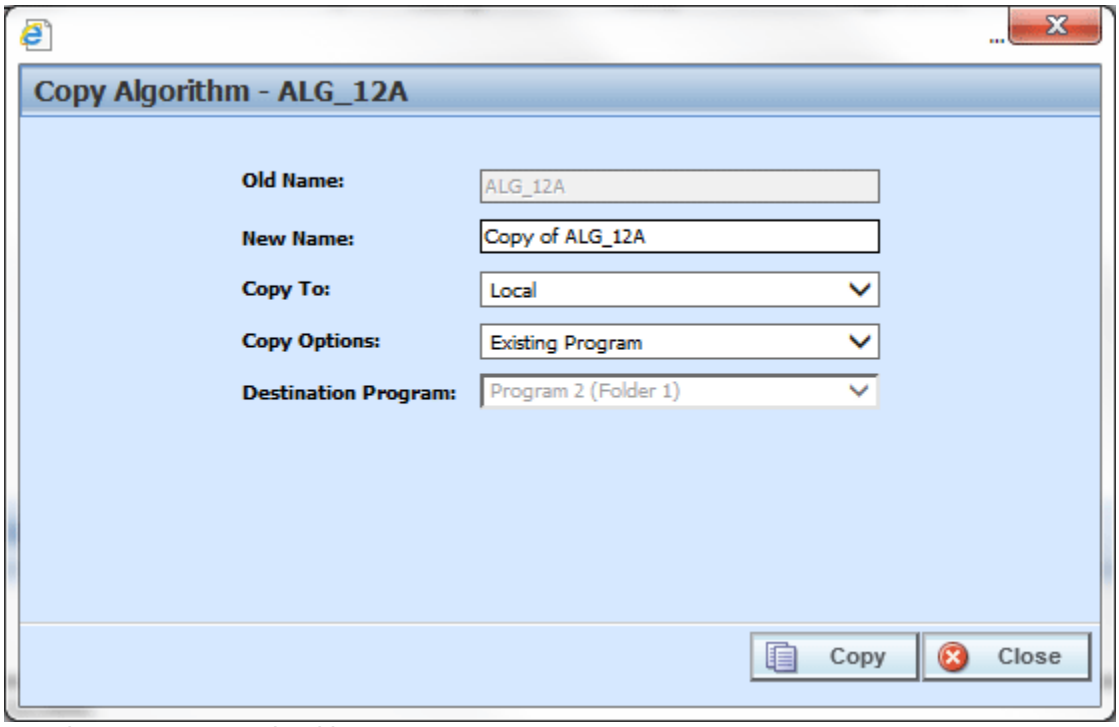


Figure 400 Copy Algorithm

3. Select the **Copy Options**. It is recommended that you change the copy options first. Changing options may reset the name field.

4. If this is a global algorithm, select if this is a local or global copy. This option is not present on local algorithms.

5. The copy options determine the Destination Program selections. If you are copying to the current program, you cannot choose the destination program. If you are copying to another program, select the **Destination Program** from the drop down menu.

6. Enter in a **New Name**. The name can be entered first. It is recommended that you change the copy options first. Changing options may reset the name field.

7. Click **Copy** to copy the algorithm. Dependencies will be searched. If there are any dependencies that need to be resolved, a window is displayed. See Resolving Dependencies.

8. If the copy was successful, you will be returned to the previous screen. Your algorithm has been copied. The last updated date is the date the copy was made.

DELETING AN ALGORITHM

When an algorithm is no longer needed, it can either be left in RateManager or deleted. When you delete an algorithm:

- The algorithm gets deleted from all versions of the program.

- All revisions of that algorithm get deleted as well.

If you are uncertain as to whether the algorithm will be used again, leave it in RateManager. This will save time if the algorithm is needed again and leaving it in RateManager will not slow down rating.
If you try to delete an algorithm that is still being used by a result group or another algorithm, you will receive an error. Algorithms that are locked or that are template generated cannot be deleted.

NOTE: *Algorithms that are in use cannot be deleted.*

Any unlocked global algorithm that is not in use can be deleted. Global algorithms that are in use, locked, or template generated cannot be deleted. If you try to delete a global algorithm that is unavailable for deletion, an error message is displayed informing you where the global algorithm is being used.

A lock icon indicates that this algorithm is locked by either a program or a template and is not open for editing or deleting.

Global algorithms can be deleted from the globals tab or from the program details tab.

To Delete an Algorithm

1. Navigate to the **Program Details tab** for the program that contains the algorithm you want to delete.

2. Select the algorithm you want to delete and right click. Select **Delete** from the menu.

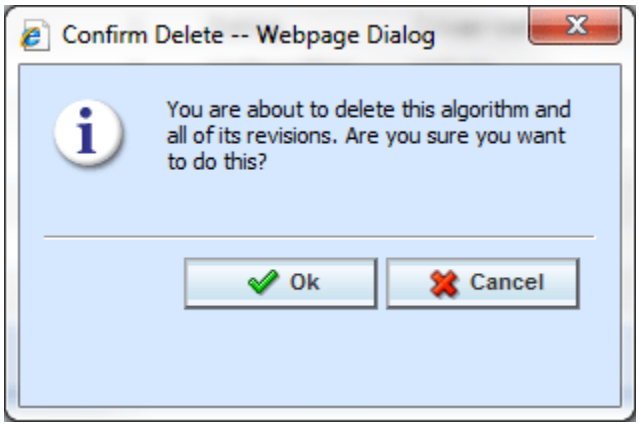


Figure 401 Deleting an Algorithm

3. You are asked to confirm the deletion of the algorithm. Select **Ok** to delete the algorithm and refresh the list of algorithms. Select **Cancel** to return to the Program Details tab.

NOTE: *If you try to delete an algorithm that is still being used in Sequencing by any program version, you will receive an error message telling you where the algorithm is being used.*

Chapter 20

DRIVER ASSIGNMENT

Driver assignment is an auto insurance specific method of assigning a specific driver to a specific vehicle on a policy, based on certain criteria that can differ by carrier basis and sometimes even by program level. The Driver Assignment option is available in the Auto product only. No other product has this option. Driver assignment is only at the global or local level. The Universals tab does not have a driver assignment option.

There are two kinds of driver assignments:

- **Scenarios**: Created under the DA scenario option using *vehicle usage options*. Vehicle usage options are specific usage sets within RateManager that allow you to build a sequence of events within a defined scenario to obtain a desired result. See Vehicle Usage Options for detailed explanations. You cannot use categories in a scenario. There are four vehicle usage options that you can use:
  - Vehicle Usage by Vehicle Order (Input:299)
  - Vehicle Usage on Drivers
  - Driver Usage on Vehicle
  - Vehicle Usage by Vehicle ID (Input:299/368)
- **Algorithms**: These algorithms work exactly the same way as any other algorithm in RateManager and allow you to create a set of rules to be executed in sequence that will obtain a desired result. Algorithms use categories and not vehicle usage options. There are four algorithm options:
  - Flag Driver
  - Flag Vehicle
  - Rank Driver
  - Rank Vehicle

Global and Local Driver Assignments

Driver Assignment Scenarios and Driver Assignment Algorithms can be created and managed at the global level and at the local level only. Global driver assignments are like program level driver assignments in the way they function and in the way they are created. In global driver assignments, you can create driver assignment scenarios and algorithms, exactly like you can at the program level. Global driver assignments are available only to programs in the auto product, same as the program level.

There are a couple of differences between globals and locals. One difference between them is that global driver assignments are available to all programs within the product while program level driver assignments are available only to the program where they were created. Another is that Global driver assignments can use only global elements. Local driver assignments can use either local or global elements.

Global: Global variables are created and managed at the global level and are product specific. Global variables can be accessed from the Global tab or from the Program Details tab.

Local: Local variables are created and managed at the program version level and are specific to the program. Local variables can be accessed from the Program Details tab.

Both global and local driver assignments can be created by the user or brought in from a template. Driver assignments that are user created are open for editing, copying, creating new revisions and deletion if they are not locked.

Driver assignments that were brought in from a template are locked at all times and cannot be unlocked. No editing or deleting is allowed but they can be copied. This is to ensure the integrity of the template. New driver assignments can be added to a template and new revisions can be created. These new driver assignments are open for editing, copying, creating new revisions or deletion, unless the program version is locked.

NOTE: *Driver Assignments are not available at the universal level.*

Separating Driver and Vehicle Output (Auto Product Only)

By default, once Driver-Assignment is complete, there are no longer drivers and vehicles there are only driver-vehicle pairs.

The advantages to this are:

1. You can use driver and vehicle level elements (fields, variables and results) inside of variables and algorithms, without having to specify which driver or vehicle to use.

   For example, if you have a class factor that depends on the assigned driver's age, as well as the vehicles use code, you would set up the class factor as a driver-level variable. SoftRater would then automatically use the correct driver and vehicle information.

2. The result XML will show all drivers, vehicle and driver-vehicle elements together. For example, if you output the driver age, it would show up under the vehicle that driver is assigned to.

The disadvantage is that once Driver-Assignment is complete, you can no longer use any elements for drivers that did not get assigned to a vehicle. For example, if driver 3 has 3 points, but did not get assigned to a vehicle, then after Driver-Assignment, those 3 points will no longer be available. While it is possible to account for those 3 points before Driver-Assignment, it often gets messy and difficult to maintain.

To remedy this, RateManager allows the user to instruct SoftRater to separate the driver and vehicle elements, while keeping the association between them. This keeps the first advantage shown above and fixes the disadvantage. However, in doing this, the driver and vehicle elements no longer show together on the result XML. Since they do not show together, the user has no way of knowing which driver got assigned to which vehicle, unless additional information is returned. You can return that additional information by following the steps below.

How to Identify the Vehicle a Driver is Assigned to and Vice Versa

1. Start by creating result variables to hold the information. You will probably want the following results:

Result Variable Name	Working Category	Type	Description
Assigned Vehicle Number	Driver	Integer	Holds the number of the vehicle that the current driver was assigned to
Assigned Driver Number	Vehicle	String	Holds the number of the driver that was assigned to the current vehicle

2. Next, create an algorithm to set the result variables.

Algorithm Name	Working Category	Description
Get Assignment Information	Driver-Vehicle	Sets the result variables Assigned Vehicle Number and Assigned Driver Number

Step Number	Type	Step
Step 1	Arithmetic	Assigned Vehicle Number (driver result) = VehicleID (vehicle field)
Step 2	String Addition	Assigned Driver Number (vehicle result) = DriverID (driver field)

3. Finally, add the new algorithm to the Sequencing, after the Driver-Assignment scenario.

Order	Name
1	abc
2	def
…	…
n	Driver Assignment
n + 1	Get Assignment Information
…	…

DRIVER ASSIGNMENT OPTIONS

The **Program Details tab** and the **Globals tab** both display all driver assignment scenarios, flag driver, flag vehicle, rank driver and rank vehicle algorithms that have been created at the respective level. In addition, the revision number, data type and working category of the scenarios and algorithms will be displayed. From this tab you will be able to:

- Creating a Driver Assignment
- Editing an Driver Assignment
- Copying an Driver Assignment
- Deleting an Driver Assignment
- Creating a new Driver Assignment Revision
- Changing the Active Revision of an Driver Assignment
- Deleting an Driver Assignment Revision

Navigating to Driver Assignment

There are two paths to view and manage driver assignments, either from the Globals tab or from the Program Details tab. The Global tab displays global driver assignments only. You cannot view or manage local driver assignments on the Global tab. On the Program Details tab, you can select to view either global driver assignments or local driver assignments. If you want to view and manage local driver assignments, you must use the Program Details tab. The views and options are the same on either tab.

From either tab, when you select Driver Assignments in the Program Details Folder, the current driver assignments will be listed in the Folder Details area.

Driver Assignments is not an option on the Program Explorer tab.

1. Modules→Program Explorer to open the Program Explorer tab.

2. Select the domain type→domain→project→product→folder→program→program version. You also can access driver assignments from the globals tab. Domain type→domain→project→ product→globals.

3. Open the program version to get the Program Details tab. Driver assignments are available from the program details folders area on the left side of the tab.

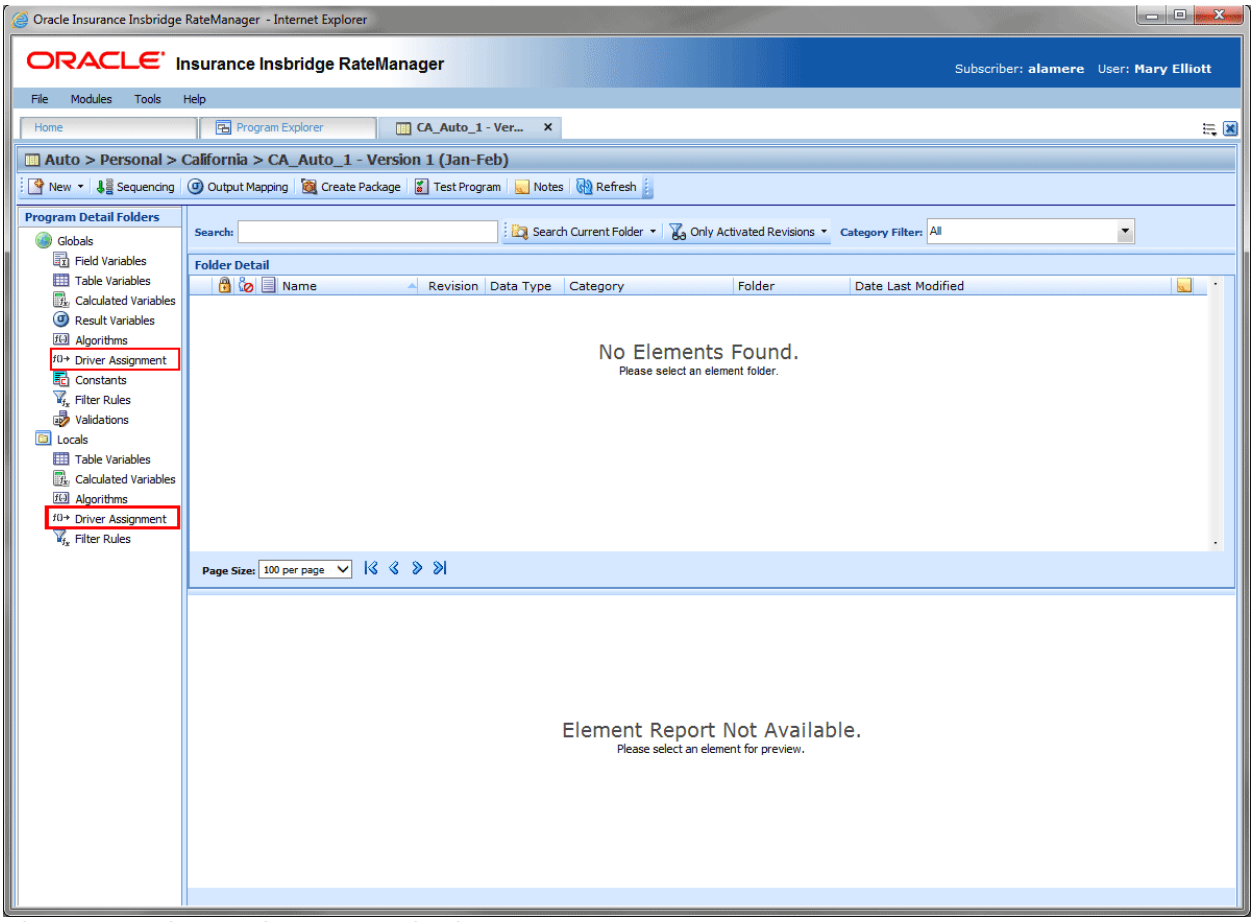


Figure 402 Driver Assignment Navigation

DRIVER ASSIGNMENT FUNCTIONALITY PROGRAM DETAILS

Available functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any unlocked driver assignment.

- **Driver Assignment Functions:** The copy and delete functions can be performed on the Program Details tab using a right click menu. Reports also can be generated using the right click menu on the Program Details tab.
- **Driver Assignment Revision Functions:** You will be able to view, copy, and delete driver assignment revisions. Active revisions can be set on the Program Details tab as well.
- **Driver Assignment Tab:** When you create a new driver assignment, edit an existing driver assignment, or work with revisions, you will be placed on the Driver Assignment tab.

The functions you can perform depend on the status of the driver assignment. Locked driver assignments will have fewer options. There are icons that can tell you at a glance what the status of the driver assignment is.

Icon	What it Indicates
	Driver Assignment. Global Driver Assignments are product specific and can be created at the global level, or from a program version. Local Driver Assignments are program specific and can be created at the program version level. Local Driver Assignments are displayed on the Program Details→Local Driver Assignments. The same driver assignments listing is also available for →Globals tab. The same options, new, edit, copy and delete are available on either tab.
	Locked Driver Assignment. Locked Driver Assignments are not open for editing. A lock icon indicates that this driver assignment is locked by a program and is not open for copying, editing or deleting. If the driver assignment is locked because the program is locked, you must return to the program and unlock it. This unlocks the driver assignment and changes can be made.
	Template Lock. If the driver assignment is locked because it came from a template, you will not be able to unlock the driver assignment. You can create a new driver assignment if necessary.
	In-Use. Driver Assignments that are in use are currently being worked on by another user in the system or driver assignments that you have open in another tab are not open for editing, copying, or deleting. The driver assignment will be available when the user exits the variable.
	Active Revision. This driver assignment is the current active revision.

Driver Assignment Screen Details

The screen details are the same for the global tab and program details. For more on screen details descriptions, please see Program Details→Screen Details.

DRIVER ASSIGNMENT TAB

The Driver Assignment tab is available when you create a new driver assignment, edit an existing driver assignment, or work with revisions. You also can be placed directly on the Driver Assignments tab from the *My Recently Opened DA Scenarios* snap on your Home tab.

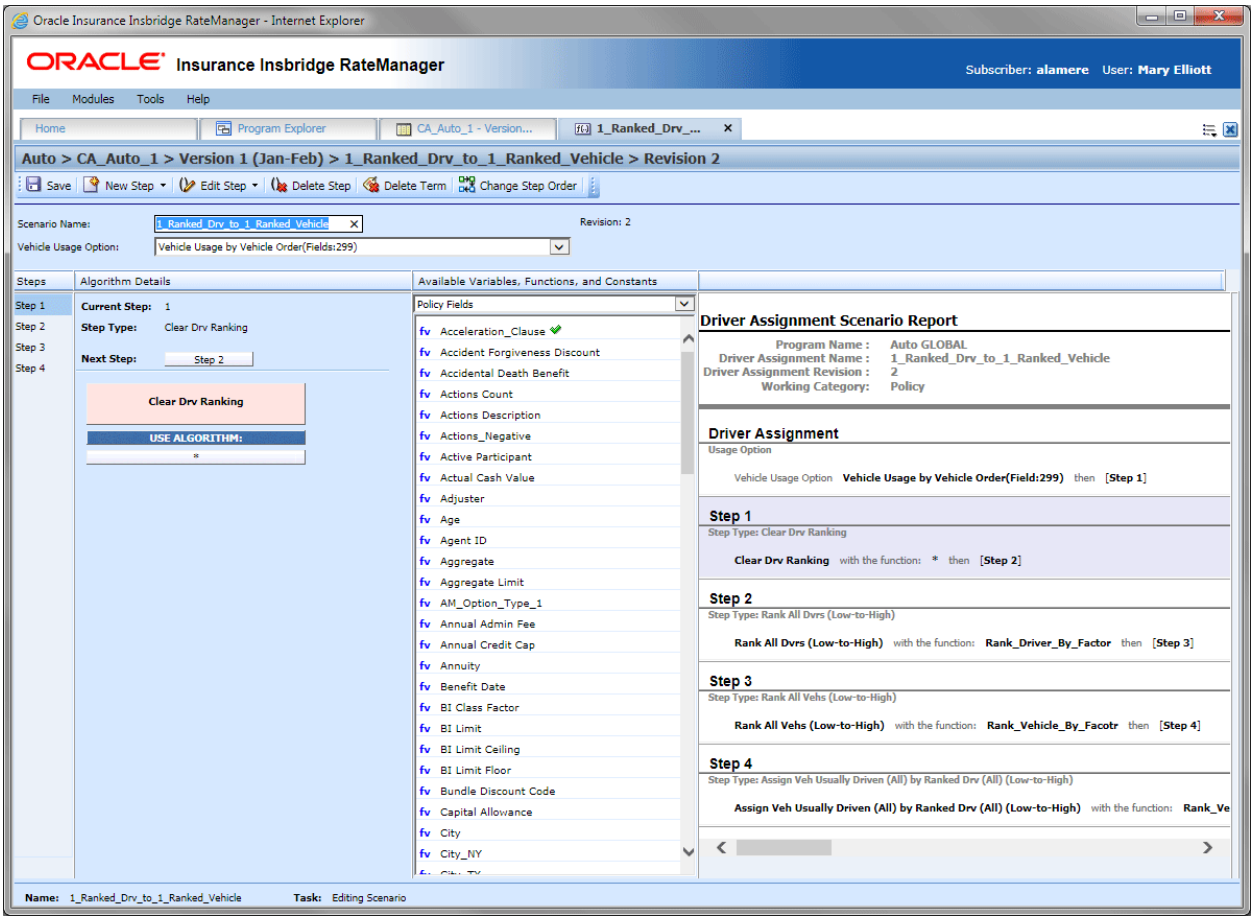


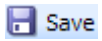
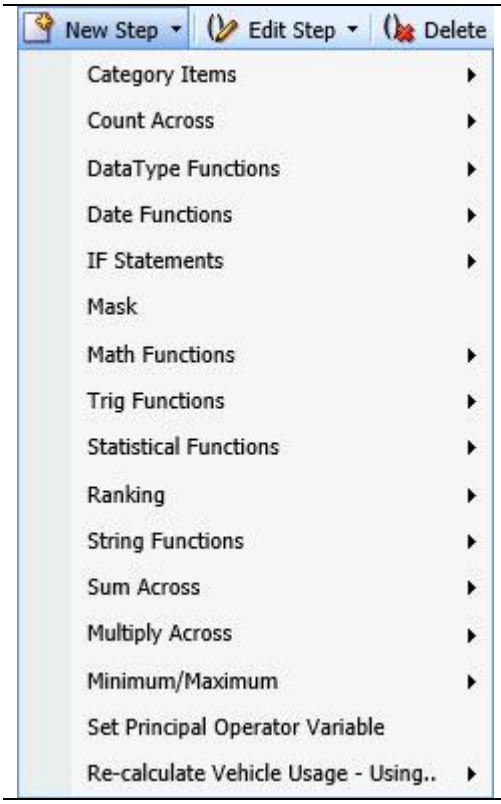
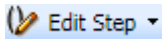
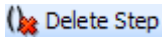
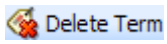
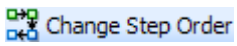
Figure 403 Driver Assignment Tab

Driver Assignment Rules

- The first step of a new driver assignment is always an arithmetic step. If you do not want an arithmetic step as your first step, add the step type you do want as Step 2. You can then delete the automatic arithmetic step.
- You cannot have a drivers assignment without any steps. There must always be at least one step. You cannot delete the last remaining step.

DRIVER ASSIGNMENT TAB FUNCTIONALITY

The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform on this tab.

	**Save:** Saves the driver assignment.
	**New Step:** Creates a new step in the driver assignment. For more information, see Working with Steps. **NOTE:** *Only Driver Assignment Scenarios have a "Set Principal Operator Variable" and a "Re-calculate Vehicle Usage: Using" step type in the list.*
	**Edit Step:** Copies the selected step or pastes the steps on the clipboard in the selected step order
	**Delete Step:** Deletes the currently selected step. You cannot delete the last step of a driver assignment.
	**Delete Term:** Deletes the currently selected term from a step in the driver assignment.
	**Change Step Order:** Allows users to change the order of the steps without using the drop and drag feature.

DRIVER ASSIGNMENT SCREEN DETAILS

There are four areas on the Driver Assignment Tab.

- **Driver Assignment Information:** For a driver assignment scenario, the driver assignment information area contains the name and defines the Vehicle Usage Option of the driver assignment. For a driver assignment algorithm, the driver assignment information area contains the name and defines the working category of the driver assignment.
- **Driver Assignment Details:** The details area is where you enter the steps that the driver assignment will perform.
- **Available Variables, Functions and Constants:** The list of the available variables, functions and constants that can be used in the Algorithm Details area.
- **Driver Assignment Report:** A listing of all the steps currently in the driver assignment.

Driver Assignment Tab Information

The driver assignment information area is where you define the driver assignment.

Naming standards can be applied to system elements. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the element and for case enforcement to be applied to any newly created element or any existing elements that are being edited. If a naming standard is in place, the pattern is displayed in the element name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented for all elements.

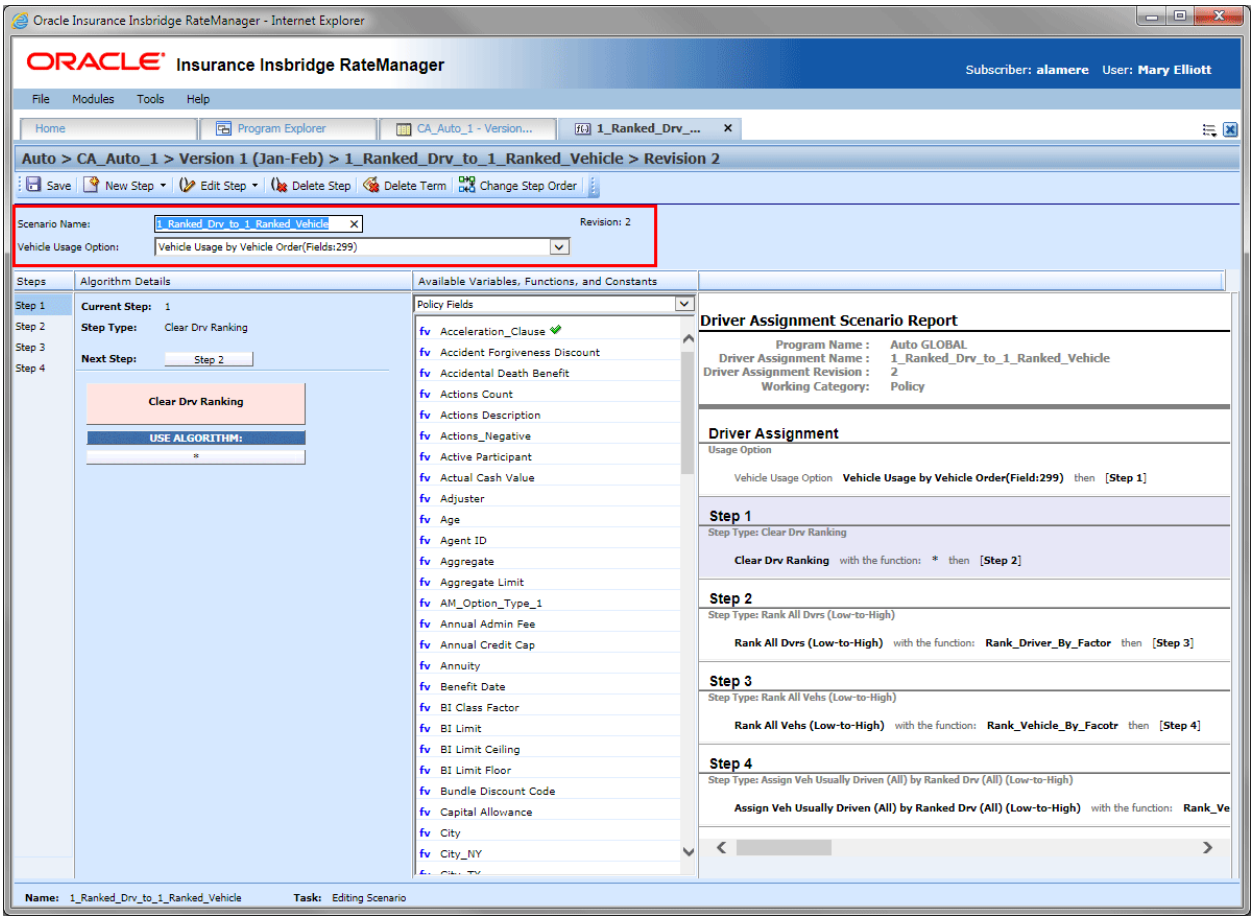


Figure 404 Driver Assignment Information Field

Screen Details

Scenario Name/Algorithm Name: Entry for what you want the driver assignment to be called. Alphanumeric characters are permitted, but no special characters (ex: &, *, ", +, @, etc.) are allowed. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

For Driver Assignment Algorithms: Working Category: Sets the working category where the driver assignment algorithm will be run. You can select the working category from the drop down menu or you can browse for a working category by clicking **BROWSE**. This brings up the Browse

Category window. From here you can navigate to the Category you want. Once you have selected the category, click OK. The category auto fills in the working category section.

The Working Category determines the options displayed in the Available Algorithms area. For example, if you select Policy as the Working Category, then some of the menu choices in the Available Variables area are DA Results, Message Templates, Constants, Step Variables, Policy Fields, Policy Variables and Policy Results.

All driver assignment scenarios are under the Policy working category.

Driver Assignment Algorithms are used in Driver Assignment scenarios only. They are not available to any other element in RateManager.

For Driver Assignment Scenario: Set Vehicle Usage Options: Set the Vehicle Usage Options where the driver assignment scenario will be run. Driver Assignment Algorithms displays the category where they were created.

Vehicle Usage Options

Vehicle Usage Options are available when you create a Drivers Assignment Scenario. Many carriers make decisions in driver assignment based on vehicle usage. The Vehicle Usage option you select will apply to the entire scenario. If you want to change vehicle usage, you must use the re-calculate vehicle usage step. In RateManager, four vehicle usage options are supported:

- Vehicle Usage by Vehicle Order (Field:299)
- Vehicle Usage on Drivers
- Driver Usage on Vehicles
- Vehicle Usage by VehicleID (Fields:299/368)

Vehicle Usage by Vehicle Order (Field:299)

Vehicle usage is designated as the actual vehicle a driver principally operates, based on the order of the vehicles in the field file. For example, driver 1 principally operates the second vehicle on the policy. In RateManager, this is accomplished using the Driver field 299 (VehPrincipallyDriven). Specify the vehicle a driver principally operates by setting the VehPrincipallyDriven field equal to the order the vehicle appears in the field file.

Vehicle Usage on Drivers

Vehicle usage is designated as a percentage a driver operates each vehicle. For example, driver 1 operates vehicle 1 50% of the time and vehicle 2 50% of the time. In RateManager, this is accomplished using the Driver fields 1-16, 50, and 700 (DrvUsePctOnVehX, where X represents an integer 1-18). This method supports an unlimited number of drivers and up to 18 vehicles. The percentages should add up to 100%.

Driver Usage on Vehicles

Vehicle usage is designated as a percentage a vehicle is operated by each driver.
For example, vehicle 1 is driven by driver 1 50% of the time and driver 2 50% of the time. In RateManager, this is accomplished using the Vehicle fields 1-16, 50, and 700 (VehUsePctOnDrvX, where X represents an integer 1-18). This method supports an unlimited number of vehicles and up to 18 drivers. The percentages should add up to 100%.

Vehicle Usage by VehicleID (Fields:299/368)

Vehicle usage is designated as the actual vehicle a driver principally operates, based on the vehicle number specified in the field file. For example, driver 1 principally operates vehicle 3. In RateManager, this is accomplished using the Vehicle field 368 (VehicleID) and Driver field 299 (VehPrincipallyDriven). Specify the vehicle a driver principally operates by setting the VehPrincipallyDriven field equal to the VehicleID of the vehicle the driver principally operates.

Driver Assignment Tab Algorithm Details

This area contains the steps that the driver assignment will perform.

A new driver assignment always begins with an arithmetic step as **Step 1**. If you do not want an arithmetic step as your first step, add the desired step and then delete the arithmetic step. To do so, highlight **Step 1** and click **Delete Step** in the menu bar at the top of the tab.

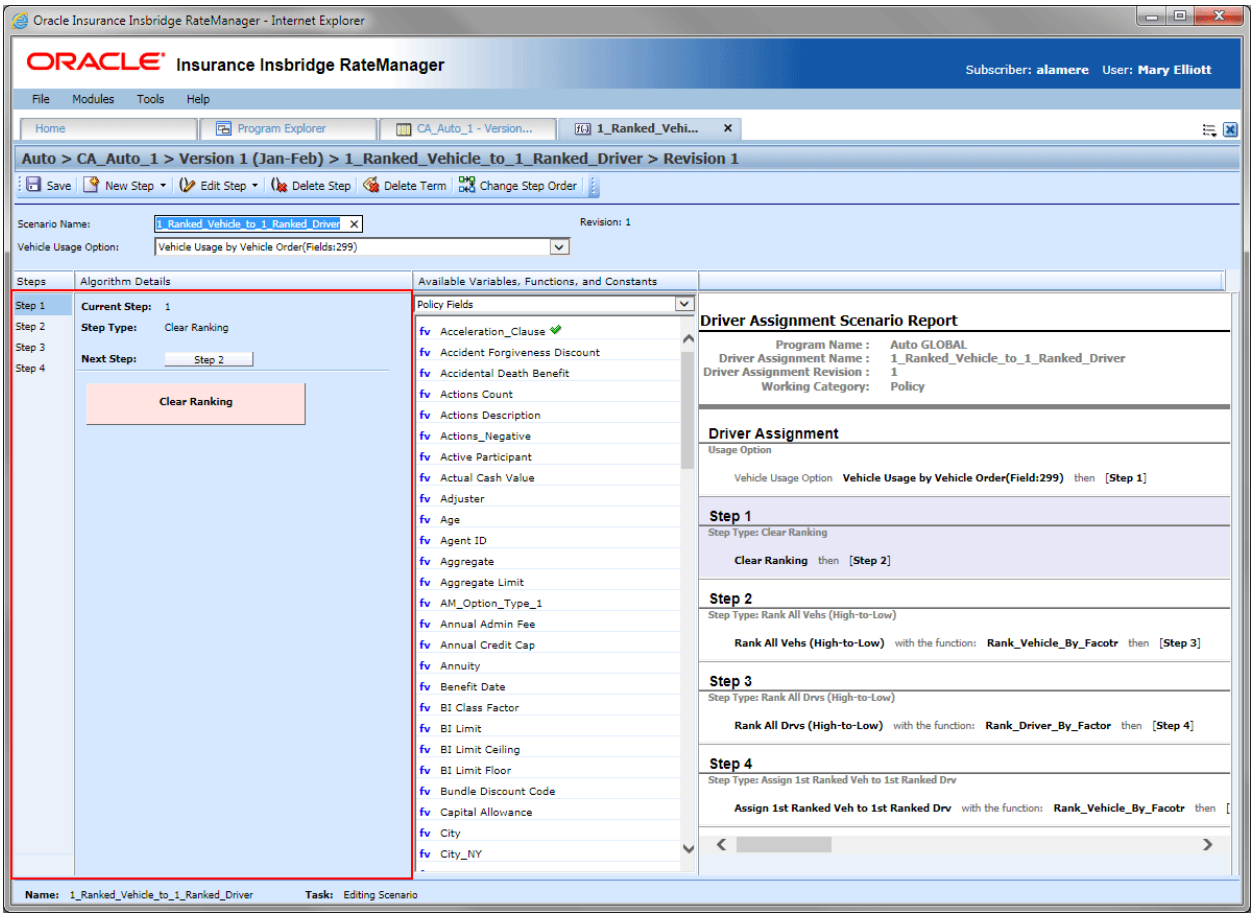


Figure 405 Algorithm Step Detail

Screen Details

Steps: Sequential order of steps created with drag-and-drop capabilities for moving a selected step up or down in the sequence.

Algorithm Details: Shows the selected step number, step type selected, next step to follow and fields/variables used for calculation. Rounding is set by clicking on the text box next to **Decimal Precision**. You can select a rounding precision from 0 to 10 digits.

Fields and variables will have a right click menu available; View Report, View Where Used and Custom Value.

Driver Assignment Tab Available Variables, Functions and Constants

The Available Variables, Functions and Constants area contains a listing of available fields, variables, and results, sorted by category; system and domain constants; and message templates that can be used as criteria. For global algorithms, only global variables or fields may be used. You cannot use any local variables in the creation of a global algorithm. A globe icon is used to indicate a global algorithm.

For Fields, you will be able to add a new field variable without leaving the Edit Driver Assignment tab. Scroll to the bottom of the Available Variables, Functions and Constants list and select <New Field Variable>. A window is displayed that allows you to enter a new field variable.

You also can add new a new result variable without leaving the Edit Driver Assignment tab.

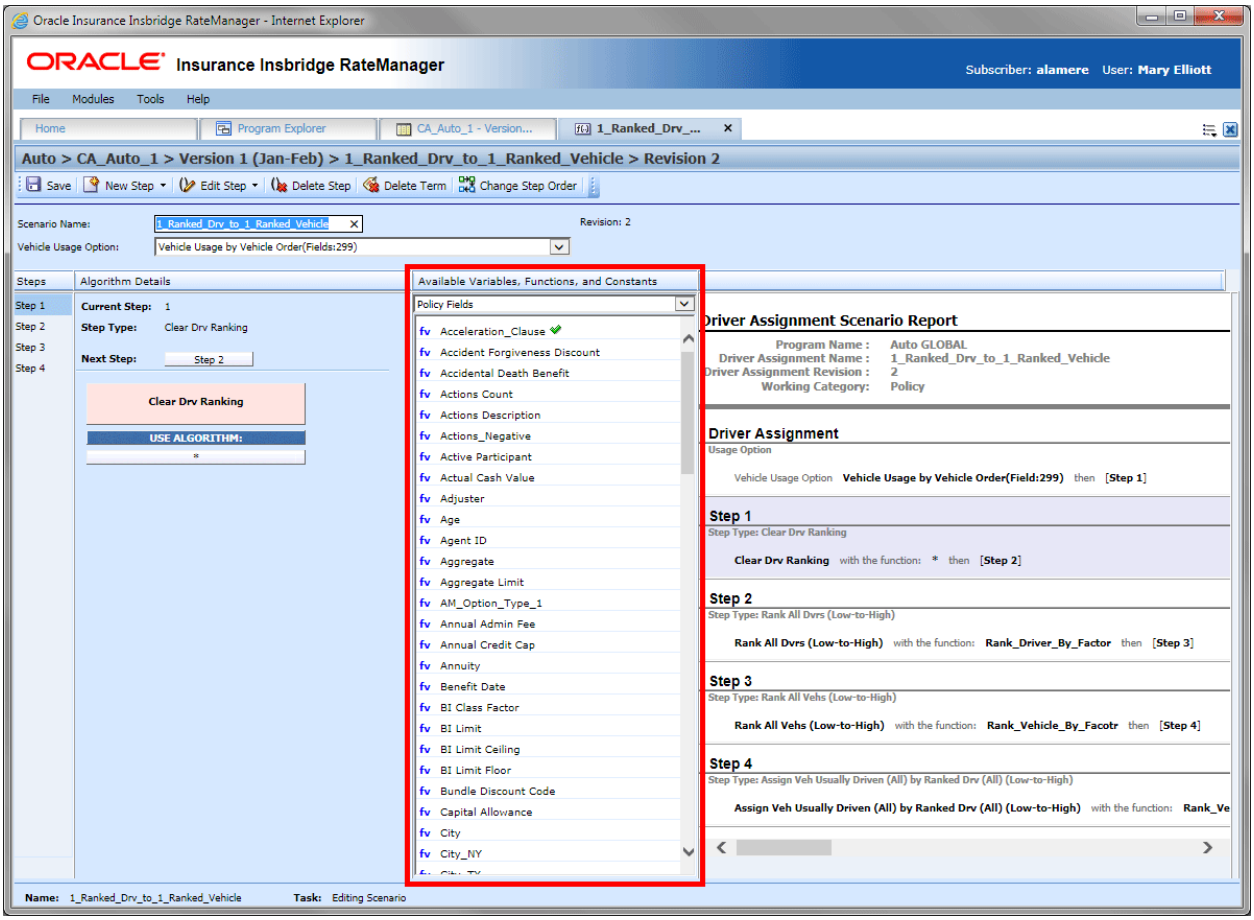


Figure 406 Available Variables, Function and Constants Field

Screen Details

Menu: The menu is the top most field. Click the down arrow to select the category you want. Policy variables will be available for all categories. Driver and Vehicle categories will be available in the Driver-Vehicle category. Categories are determined in the Algorithm Information area.

Selection: The available variables, functions, constants and step variables will be listed. Variables will be based on the type of driver assignment you select.

Selections for Available Variables, Functions and Constants:

The selections in the Available Variables, Functions and Constants menu depends upon the type of driver assignment you have selected. Some selection will be available in both driver assignment algorithms and driver assignment scenarios. Some selections, such as Algorithms and Functions, are available in driver assignment scenarios only.

Selection	Description
Functions: Global Versioning	Select the program with the global versioning you want to use.
Functions: Library Callouts	Select a SoftLibrary that has been registered in the IBFA. For callouts, step types do not matter. The entire step will consist of the callout. Callouts are program specific and are available in local algorithms and driver assignments only.
Functions: Program Callouts	Select the program to be called out to. For callouts, step types do not matter. The entire step will consist of the callout. Callouts are program specific and are available in local algorithms and driver assignments only.
DA Driver Results	System generated result variables that can be used to hold results for drivers. **Available in Driver Assignment scenarios only.**
DA Vehicle Results	System generated result variables that can be used to hold results for vehicles. **Available in Driver Assignment scenarios only.**
Algorithms: Flag Driver	Previously entered Driver assignment algorithms: flag driver, flag vehicle, rank driver and rank vehicle will be listed. Only driver assignment algorithms entered within the will be available. **Available in Driver Assignment scenarios only.**
Algorithms: Flag Vehicle	
Algorithms: Rank Driver	
Algorithms: Rank Vehicle	
Functions: Flagging	Driver assignment functions that perform specific actions, flagging, ranking, assignments and miscellaneous clearing functions. **Available in Driver Assignment scenarios only.** See Scenario Functions for more information.
Functions: Ranking	
Functions: Assignments	
Functions: Miscellaneous	
Message Templates	Message templates are prefabricated messages with variable placeholders that allow users to create standard messages using variable information that is generated at the time of execution.
System Level Constants	Pre-defined constants, including category instances and system date and time, and system wide user-defined constants. Or you can create a custom value for use in this program.
Domain Constants	User-defined constants found only in a selected product.
Step Variables	Results from the previously entered steps.
Policy Fields	Fields in the Policy category.
Policy Variables	Table and calculated variables in the Policy category.
Policy Results	Result variables in the Policy category.

Driver Assignment Tab Scenario or Algorithm Report

This area contains a detailed list of the steps you have entered. This information is view only. No actions can be performed in this area.

Driver assignment scenarios have a scenario report.

Driver assignment algorithms have an algorithm report.

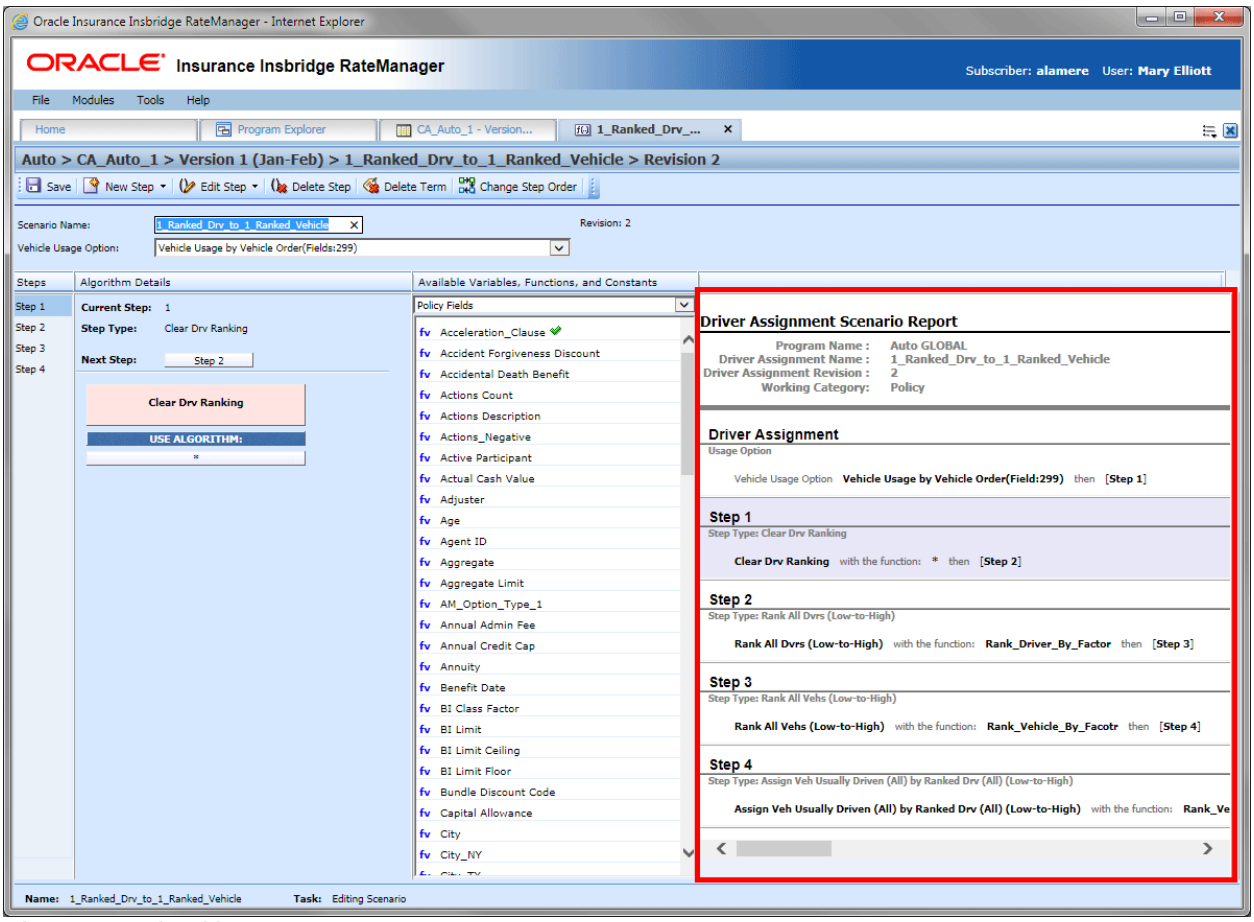


Figure 407 Algorithm Report

SCENARIO FUNCTIONS

In order to reproduce a variety of different carrier logic flows, RateManager has specific functions that are available only in a driver assignment scenario. These functions work similar to normal algorithm steps, except that the functions usually require an association with an algorithm instead of a variable. Inside a driver assignment scenario, the functions are available from the **Available Variables, Functions, and Constants** drop down.

To add one of the functions to your driver assignment scenario, double-click the one you want to add from the Available Variables, Functions, and Constants drop down menu.

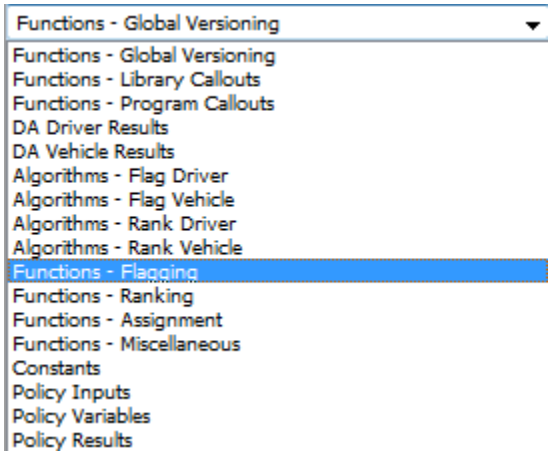


Figure 408 Functions Menu for Driver Assignment Scenario

The function categories are as follows:

- Flagging Functions
- Ranking Functions
- Assignment Functions
- Miscellaneous Functions

FLAGGING FUNCTIONS

Before covering all flagging functions available, it is important to understand the RateManager driver assignment concept of flagging. This process can be compared to that of filtering. For example, if you would like to select all drivers under the age of 21, in driver assignment you would make the statement "Flag all drivers under the age of 21". This operation is usually performed before making a calculation or assignment determination on a specific set of drivers or vehicles.

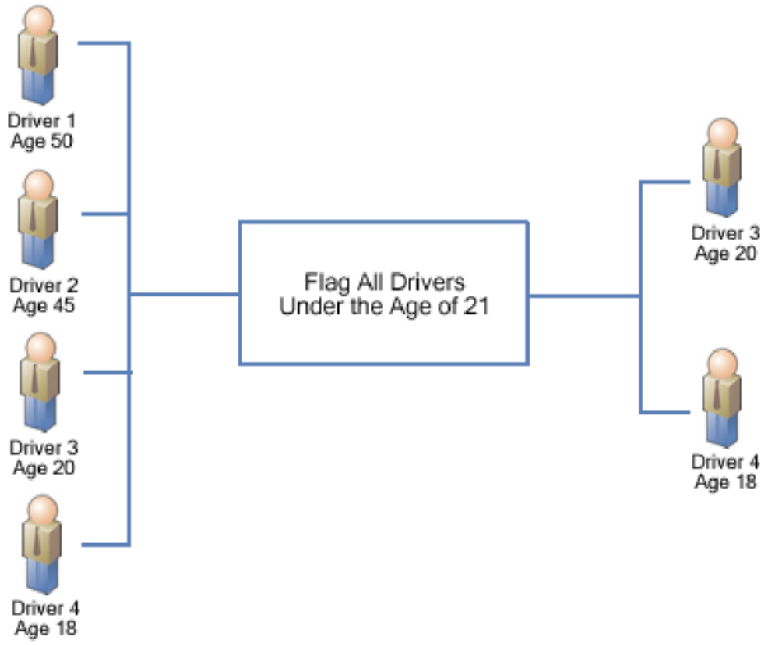
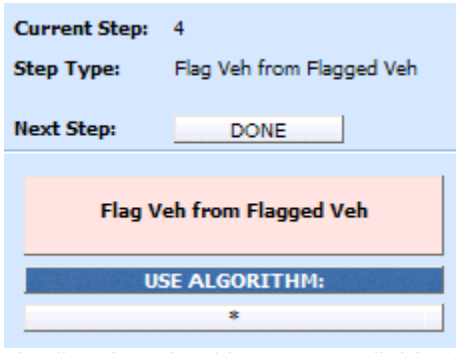


Figure 409 Flagging Functions

Most flagging functions require an association with a predefined flagging algorithm in order to determine custom criteria for the function. For more information, see Flagging Algorithms. Flagging functions are only available from within a driver assignment scenario.

To add a flagging function to your driver assignment scenario, select **Functions - Flagging** from the **Available Variables, Functions, and Constants** drop down and then double-click the flagging function you want to add. If the flagging function requires an association with a flagging algorithm, a box with an asterisk (*) is displayed in the step type. Replace the asterisk box with a flagging algorithm the same way you would replace an item in a calculated variable or algorithm step.



The flagging algorithms are available from the **Available Variables, Functions, and Constants** drop down under **Algorithms - Flag Driver** and **Algorithms - Flag Vehicle.**

Available Flagging Functions

Flag 1st Ranked Drv

This flagging operation should only be performed after a driver ranking function. It chooses the 1st or top ranked driver from the previous ranking operation. This is a standalone function, so an association with an algorithm is not required.

Flag 1st Ranked Veh

This flagging operation should only be performed after a vehicle ranking function. It chooses the 1st or top ranked vehicle from the previous ranking operation. This is a standalone function, so an association with an algorithm is not required.

Flag All Vehs

All vehicles are in the scope and available for flagging, therefore ignoring any previously flagged vehicles. An association with a predefined flag vehicle algorithm must be made in this function.

Flag Assigned Vehs

All assigned vehicles are in the scope and available for flagging, therefore ignoring any previously flagged vehicles. An association with a predefined flag vehicle algorithm must be made in this function.

Flag Drvs [All]

All drivers are in the scope and available for flagging, therefore ignoring any previously flagged drivers. An association with a predefined flag driver algorithm must be made in this function.

Flag Drvs [All] (Usage Set)

This flagging operation should only be used when the Vehicle Usage Option is set to Vehicle Usage on Drivers. This flagging operation creates a collection, or usage set, of drivers for each vehicle that can then be ranked and assigned using similar Usage Set functions. Currently, this operation will function correctly only when “Weight Usage Result” is selected in the “Use Algorithm” field. The “Weight Usage Result” selection creates usage sets of drivers consisting of any driver whose DrvUsePctOnVehX field value for the corresponding vehicle is not null.

Example 1:

Driver 1
DrvUsePctOnVeh1 = 30
DrvUsePctOnVeh2 = 70
DrvUsePctOnVeh3 = NULL

Driver 2
DrvUsePctOnVeh1 = NULL
DrvUsePctOnVeh2 = 10
DrvUsePctOnVeh3 = 90
Vehicle 1’s usage set of drivers will include Driver 1 only.
Vehicle 2’s usage set of drivers will include both Driver 1 and Driver 2.
Vehicle 3’s usage set of drivers will include Driver 2 only.

This flagging function also lets the user specify up to three vehicle-level fields or variables that can be used in a Rank Driver or Flag Driver algorithms, where vehicle-level fields or variables would ordinarily not be available. These variables are labeled “Usage Set Variable #1-3” and are available under the “DA Driver Results” available variable listing.

The “Weight Usage Result” system variable can also be used in a Rank Driver or Flag Driver algorithm, after the Flag Drvs [All] (Usage Set) operation has been performed. This system variable returns the value of the driver’s DrvUsePctOnVehX field corresponding to the vehicle usage set that a Rank Driver or Flag Driver algorithm is operating on.

Example 2:

Using the usage sets created in Example 1, if we are ranking the drivers in Vehicle 2’s usage set high-to-low based on driver use percentage, we would create a Rank Driver algorithm using the Weight Usage Result. In this case, Driver 1 would be ranked before Driver 2 because Driver 1’s DrvUsePctOnVeh2 = 70 while Driver 2’s DrvUsePctOnVeh2 = 10.

This flagging option will only be available on Vehicle Usage Options, Vehicle Usage on Drivers and Driver Usage on Vehicles.

Flag Drvs from Flag Drvs

This flagging operation should only be performed after a flag driver function. All previously flagged drivers are in the scope and available for further flagging. An association with a predefined flag driver algorithm must be made in this function.

Flag Drvs: Assigned

All assigned drivers are in the scope and available for flagging, therefore ignoring any previously flagged drivers. An association with a predefined flag driver algorithm must be made in this function.

Flag Drvs: Unassigned

All unassigned drivers are in the scope and available for flagging, therefore ignoring any previously flagged drivers. An association with a predefined flag driver algorithm must be made in this function.

Flag Last Ranked Drv

This flagging operation should only be performed after a driver ranking function. It chooses the last ranked driver from the previous ranking operation. This is a standalone function, so an association with an algorithm is not required.

Flag Last Ranked Veh

This flagging operation should only be performed after a vehicle ranking function. It chooses the last ranked vehicle from the previous ranking operation. This is a standalone function, so an association with an algorithm is not required.

Flag Unassigned Vehs

All unassigned vehicles are in the scope and available for flagging, therefore ignoring any previously flagged vehicles. An association with a predefined flag vehicle algorithm must be made

in this function.

Flag Veh from Flagged Veh

This flagging operation should only be performed after a flag vehicle function. All previously flagged vehicles are in the scope and available for further flagging. An association with a predefined flag vehicle algorithm must be made in this function.

FLAGGING ALGORITHMS

Flagging algorithms are used in flagging functions to determine which drivers or vehicles to flag. Flagging algorithms work just like calculated variables and algorithms only they are created from the Driver Assignment Listing screen.

RateManager supports two types of flagging algorithms:

- **Flag Vehicle** algorithms
- **Flag Driver** algorithms

Both work the same in that at the end of the algorithm, you set a result equal to 1 to flag the item or 0 to not flag the item. The only difference being that **Flag Vehicle** algorithms are used to flag vehicles by setting **Vehicle Result,** while **Flag Driver** algorithms are used to flag drivers by setting **Driver Result.**

Shown below is an example of a **Flag Driver** algorithm. It uses a predefined driver calculated variable, Driver Age, to determine if it should set the **Driver Result.**

Step 1: IF Driver Age [less than] 21 then [Step 2] else [Done]
Step 2: Driver Result = 1

Vehicle Result and **Driver Result** are available from the **Available Variables, Functions, and Constants** drop down under **DA Vehicle Results** and **DA Driver Results,** respectively.

RANKING FUNCTIONS

A ranking function ranks the specified items based on the specified criteria. Ranking functions are usually used prior to making an assignment determination on a specific set of drivers or vehicles.

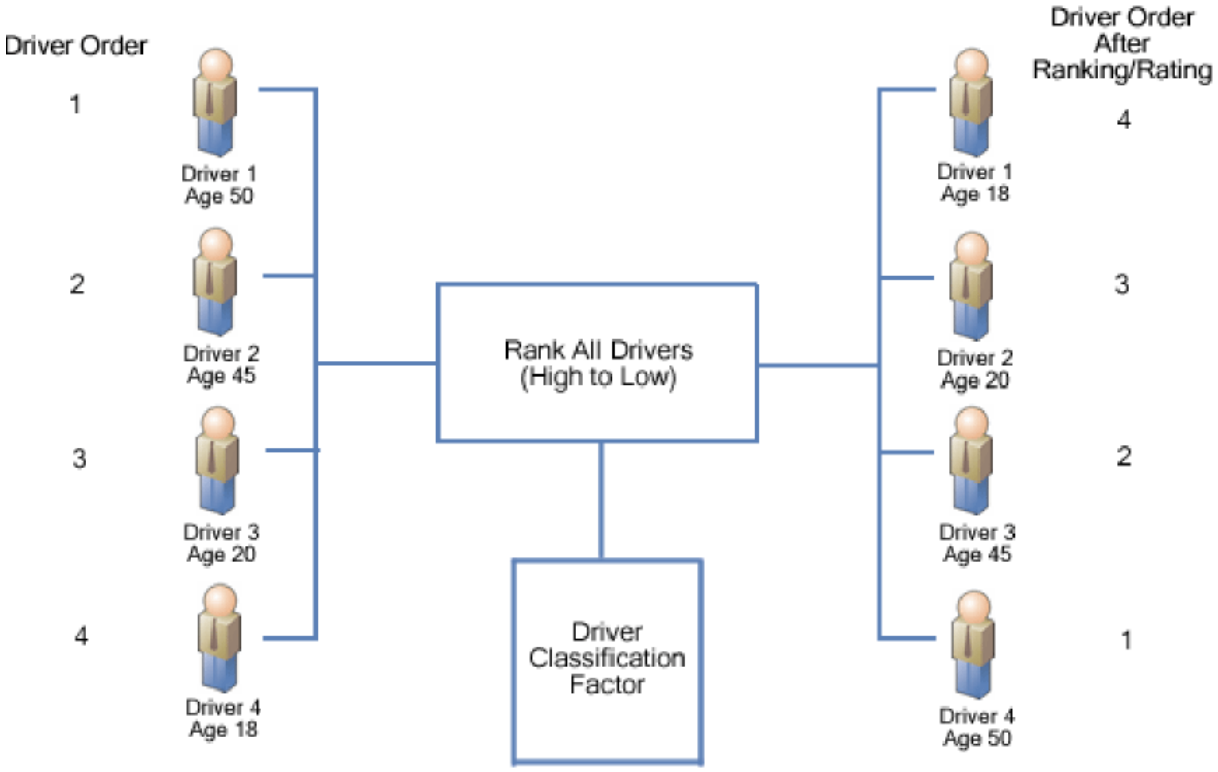
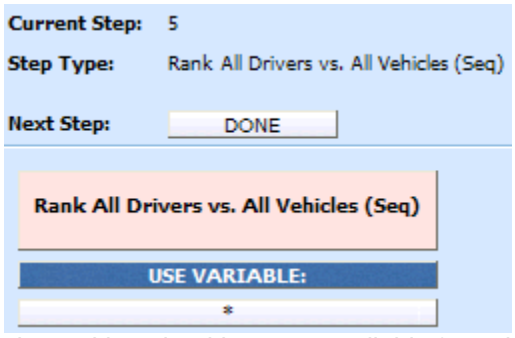


Figure 410 Ranking Functions

All ranking functions require an association with a predefined ranking algorithm, in order to determine custom criteria for ranking. For more information, see Ranking Algorithms. Ranking functions are only available from within a driver assignment scenario.

To add a ranking function to your driver assignment scenario, select **Functions - Ranking** from the **Available Variables, Functions, and Constants** drop down and then double-click the ranking function you want to add. Since the ranking function requires an association with a ranking algorithm, a box with an asterisk (*) will appear in the step type. Replace the asterisk box with a ranking algorithm the same way you would replace an item in a calculated variable or algorithm step.



The ranking algorithms are available from the **Available Variables, Functions, and Constants** drop down under **Algorithms - Rank Driver** and **Algorithms - Rank Vehicle.**

Available Ranking Functions

Rank All Drvs (Low-to-High)

This function ranks all of the drivers, according to the specified function, in ascending order (lowest to highest).

Rank All Drivers vs. All Vehicles (Seq)

This ranking function uses the result of a Driver-Vehicle variable and ranks every possible Driver & Vehicle pair, high to low. This step should be performed prior to the 'Assign 1st Ranked Veh to 1st Ranked Driver (Seq)' assignment functions.

Rank All Drvs (High-to-Low)

This function ranks all of the drivers, according to the specified function, in descending order (highest to lowest).

Rank All Drvs (High-to-Low) (Usage Set - Conditional)

This ranking operation should only be performed after a Flag Drvs [All] (Usage Set) operation. An If step is used to test criteria in the scope of the drivers in each usage set. The drivers in each usage set are then ranked, using one of two specified functions, in descending order (highest to lowest).

Rank All Drvs (High-to-Low) (Usage Set)

This ranking operation should only be performed after a Flag Drvs [All] (Usage Set) operation. This function ranks all of the drivers in each usage set, according to the specified function, in descending order (highest to lowest). This ranking operation should be used when the Vehicle Usage Option is set to Vehicle Usage on Drivers.

Rank All Drvs (Low-to-High) (Usage Set - Conditional)

This ranking operation should only be performed after a Flag Drvs [All] (Usage Set) operation. An If step is used to test criteria in the scope of the drivers in each usage set. The drivers in each usage set are then ranked, using one of two specified functions, in ascending order (lowest to highest).

Rank All Drvs (Low-to-High) (Usage Set)

This ranking operation should only be performed after a Flag Drvs [All] (Usage Set) operation. This function ranks all of the drivers in each usage set, according to the specified function, in ascending order (lowest to highest). This ranking operation should be used when the Vehicle Usage Option is set to Vehicle Usage on Drivers.

Rank All Vehs (High-to-Low)

This function ranks all of the vehicles, according to the specified function, in descending order (highest to lowest).

Rank All Vehs (Low-to-High)

This function ranks all of the vehicles, according to the specified function, in ascending order (lowest to highest).

Rank Assigned Vehs (High-to-Low)

All assigned vehicles are in the scope and available for ranking, therefore ignoring any unassigned vehicles. Using only the current scope, it ranks all of the vehicles, according to the specified function, in descending order (highest to lowest).

Rank Assigned Vehs (Low-to-High)

All assigned vehicles are in the scope and available for ranking, therefore ignoring any unassigned vehicles. Using only the current scope, it ranks all of the vehicles, according to the specified function, in ascending order (lowest to highest).

Rank Flagged Drvs (High-to-Low)

This ranking operation should always be performed after a flag driver operation. All previously flagged drivers are in the scope and available for ranking. Using only the current scope, it ranks all of the drivers, according to the specified function, in descending order (highest to lowest).

Rank Flagged Drvs (Low-to-High)

This rating/ranking operation should always be performed after a flag driver operation. All previously flagged drivers are in the scope and available for ranking. Using only the current scope, it ranks all of the drivers, according to the specified function, in ascending order (lowest to highest).

Rank Unassigned Vehs (High-to-Low)

All unassigned vehicles are in the scope and available for ranking, therefore ignoring any assigned vehicles. Using only the current scope, it ranks the vehicles, according to the specified function, in descending order (highest to lowest).

Rank Unassigned Vehs (Low-to-High)

All unassigned vehicles are in the scope and available for ranking, therefore ignoring any assigned vehicles. Using only the current scope, it ranks the vehicles, according to the specified function, in ascending order (lowest to highest).

RANKING ALGORITHMS

Ranking algorithms are used in ranking functions to determine how to rank drivers or vehicles. Ranking algorithms work just like calculated variables and algorithms, only they are created from the Driver Assignment Listing screen.

RateManager supports two types of ranking algorithms:

- **Rank Vehicle** algorithms
- **Rank Driver** algorithms

Both algorithms work the same in that at the end of the algorithm, you set a result equal to the value to be used for ranking. The only difference being that **Rank Vehicle** algorithms are used to rank vehicles by setting **Vehicle Result,** while **Rank Driver** algorithms are used to rank drivers by setting **Driver Result**. Shown below is an example of a **Rank Driver** algorithm. It uses a predefined driver table variable, Pleasure Use Factor, to set the **Driver Result**.

Step 1: **Driver Result** = Pleasure Use Factor

Vehicle Result and **Driver Result** are available from the **Available Variables, Functions, and Constants** drop down under **DA Vehicle Results** and **DA Driver Results,** respectively.

ASSIGNMENT FUNCTIONS

In driver assignment, there are many ways to assign a particular driver, or set of drivers, to a particular vehicle, or set of vehicles. SoftRater assigns drivers to vehicles using assignment functions. All assignment functions are standalone, meaning they do not require an association with a predefined algorithm.

Assignment functions are only available from within a driver assignment scenario. To add an assignment function to your driver assignment scenario, select **Functions - Assignment** from the list of **Available Variables, Functions, and Constants** drop down and then double-click the assignment function you want to add.

For all assignment functions, a vehicle cannot be reassigned, e.g. if vehicle 1 was assigned to driver 1 in step 3 of the driver assignment, vehicle 1 cannot be assigned to a different driver in step 5. Also, SoftRater will not allow you to exit driver assignment without assigning a driver to each vehicle. In the event that your driver assignment scenario is finished, and vehicles remain unassigned, SoftRater will automatically assign the first driver to each of those vehicles.

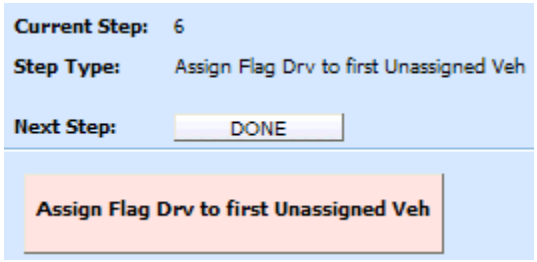


Figure 411 Adding an Assignment Function

Available Assignment Functions

Assign 1st Ranked Veh to 1st Ranked Drv

This assignment operation should only be performed after both a rank vehicle function and a rank driver function. It will assign the 1st or top ranked vehicle to the 1st or top ranked driver.

Assign 1st Ranked Veh to 1st Ranked Drv (Seq)

This assignment function should be performed after the 'Rank All Drivers vs. All Vehicles (Seq)' ranking function. The highest ranked Driver & Vehicle pairs are assigned until all vehicles have

been assigned a driver. If the Driver & Vehicle pairs are ranked such that a driver will be assigned to more than one vehicle, this assignment function will assign the driver to multiple vehicles.

Assign 1st Ranked Veh to 1st Ranked Drv (Seq) (with Exclusion)

This assignment function should be performed after the 'Rank All Drivers vs. All Vehicles (Seq) ranking function. The highest ranked Driver & Vehicle pairs are assigned until all vehicles have been assigned a driver. If the Driver & Vehicle pairs are ranked such that a driver will be assigned to more than one vehicle, this assignment function will not assign the driver to multiple vehicles, skipping the Driver & Vehicle pair until all vehicles have been assigned unique drivers.

Assign 1st Ranked Veh to 1st Ranked Drv (Usage Set)

This assignment operation should only be performed after both a Rank Vehicle function and a Rank Drivers (Usage Set) function. It will assign the 1st or top ranked vehicle to the 1st or top ranked driver in the vehicle's usage set.

This ranking option will only be available on Vehicle Usage Options, Vehicle Usage on Drivers and Driver Usage on Vehicles.

Assign 1st Ranked Veh to 1st Ranked Drv (with Exclusion)

This assignment function should be performed after a Rank Vehicle function and a Rank Driver function. The highest ranked driver will be assigned to the highest ranked vehicle, then the second-highest ranked driver will be assigned to the second-highest ranked vehicle, and so on. Each driver will be assigned to only one vehicle in this assignment function. An additional assignment function will need to be included to handle unassigned vehicles.

Assign 1st Ranked Veh to Flag Drv

This assignment operation should only be performed after both a rank vehicle and flag driver function. It will assign the 1st or top ranked vehicle to the flagged driver. If multiple drivers are flagged, the first driver that was flagged will be assigned to the 1st ranked vehicle.

Assign All Vehs by Princ Op (Drvs NONEXCLUSIVE)

This assignment operation is performed after the **Set Principal Operator Variable** function, unless the principal operator indicator is passed as (field ID 253). This assignment operation assigns all drivers to the vehicle they principally drive. The assignment will be performed nonexclusively, meaning that if a driver is the principal operator on more than one vehicle, the driver will be assigned to each one.

Assign Flag Drv to all Unassigned Vehs

This assignment operation should only be performed after a flag driver function. It will assign the currently flagged driver to all unassigned vehicles. If multiple drivers are flagged, the first driver that was flagged will be assigned to all unassigned vehicles.

NOTE: *This function is usually used in left over vehicle situations.*

Assign Flag Drv to first Unassigned Veh

This assignment operation should only be performed after a flag driver function. It will assign the currently flagged driver to all unassigned vehicles. If multiple drivers are flagged, the first driver that was flagged will be assigned to all unassigned vehicles.

NOTE: *This function is usually used in left over vehicle situations.*

Assign Unassigned Vehs by Princ Op (Drvs EXCLUSIVE)

This assignment operation is performed after the **Set Principal Operator Variable** function, unless the principal operator indicator is passed as (field ID 253). This assignment operation assigns all unassigned vehicles to the driver that principally operates them. The assignment will be performed exclusively, meaning that if a driver is the principal operator on more than one vehicle, the driver will be assigned to only the first vehicle they principally operate, and if that vehicle is already assigned, the driver will not be assigned in this step.

Assign Unassigned Vehs by Princ Op (Drvs NONEXCLUSIVE)

This assignment operation is performed after the **Set Principal Operator Variable** function, unless the principal operator indicator is passed as (field ID 253). This assignment operation assigns all unassigned vehicles to the driver that principally operates them. The assignment will be performed nonexclusively, meaning that if a driver is the principal operator on more than one vehicle, the driver will be assigned to each vehicle they principally operate.

Assign Veh Usually Driven (All) by Flag Drv

This assignment operation should only be performed after a flag driver function. It will assign the flagged drivers to the vehicle(s) they operate most frequently. In the event a driver operates more than one vehicle the same amount of time, the driver will be assigned to each of those vehicles. If multiple flagged drivers principally operate the same vehicle, the driver that appears first in the field file will be assigned to the vehicle.

Assign Veh Usually Driven (All) by Ranked Drv (All) (High-to-Low)

This assignment operation should only be performed after a rank driver function. It will assign all ranked drivers to all vehicle(s) they operate most frequently. Assignment will start with the highest ranked driver and work down to the lowest ranked driver. If multiple ranked drivers principally operate the same vehicle, the driver with the highest ranking will be assigned to the vehicle. In the event a driver operates more than one vehicle the same amount of time, the driver will be assigned to each of those vehicles. The set of driver/vehicle usage fields used is determined by the Vehicle Usage Option selected on the Driver Assignment Scenario.

Assign Veh Usually Driven (All) by Ranked Drv (All) (Low-to-High)

This assignment operation should only be performed after a rank driver function. It will assign all ranked drivers to all vehicle(s) they operate most frequently. Assignment will start with the highest ranked driver and work down to the lowest ranked driver. If multiple ranked drivers principally operate the same vehicle, the driver with the highest ranking will be assigned to the vehicle. In the event a driver operates more than one vehicle the same amount of time, the driver will be assigned to each of those vehicles. The set of driver/vehicle usage fields used is determined by the Vehicle Usage Option selected on the Driver Assignment Scenario.

Assign Veh Usually Driven (All) by Ranked Drv (High-to-Low)

This assignment operation should only be performed after a rank driver function. It will assign the ranked drivers to the vehicle(s) they operate most frequently. Assignment will start with the highest ranked driver and work down to the lowest ranked driver. If multiple ranked drivers principally operate the same vehicle, the driver with the highest ranking will be assigned to the vehicle. In the event a driver operates more than one vehicle the same amount of time, the driver will be assigned to each of those vehicles.

Assign Veh Usually Driven (All) by Ranked Drv (Low-to-High)

This assignment operation should only be performed after a rank driver function. It will assign the ranked drivers to the vehicle(s) they operate most frequently. Assignment will start with the lowest ranked driver and work up to the highest ranked driver. If multiple ranked drivers principally operate the same vehicle, the driver with the highest ranking will be assigned to the vehicle. In the event a driver operates more than one vehicle the same amount of time, the driver will be assigned to each of those vehicles.

Assign Veh Usually Driven by Flag Drv

This assignment operation should only be performed after a flag driver function. It will assign the flagged drivers to the vehicle they operate most frequently. In the event a driver operates more than one vehicle the same amount of time, the driver will not be assigned to any vehicles. If multiple flagged drivers principally operate the same vehicle, the driver that appears first in the field file will be assigned to the vehicle.

Assign Veh Usually Driven by Ranked Drv (High-to-Low)

This assignment operation should only be performed after a rank driver function. It will assign the ranked drivers to the vehicle they operate most frequently. Assignment will start with the highest ranked driver and work down to the lowest ranked driver. If multiple ranked drivers principally operate the same vehicle, the driver with the highest ranking will be assigned to the vehicle. In the event a driver operates more than one vehicle the same amount of time, the driver will not be assigned to any vehicles.

Assign Veh Usually Driven by Ranked Drv (Low-to-High)

This assignment operation should only be performed after a rank driver function. It will assign the ranked drivers to the vehicle they operate most frequently. Assignment will start with the lowest ranked driver and work up to the highest ranked driver. If multiple ranked drivers principally operate the same vehicle, the driver with the lowest ranking will be assigned to the vehicle. In the event a driver operates more than one vehicle the same amount of time, the driver will not be assigned to any vehicles.

Assign Veh Usually Driven by Unassigned Drvs

This assignment operation will assign drivers that have not been assigned, to the vehicle they operate most frequently. If multiple unassigned drivers principally operate the same vehicle, the driver that appears first in the field file will be assigned to the vehicle.

Assign Vehs by Drv Usage using Last Ranked Veh

This assignment operation should only be performed when the Vehicle Usage Option is set to **Driver Usage on Vehicles.** It also should only be performed after a rank vehicle function. It will assign vehicles, starting with the last ranked vehicle, to the driver that operates the vehicle most frequently.

Assign Vehs by Drv Usage using 1st Ranked Veh

This assignment operation should only be performed when the Vehicle Usage Option is set to **Driver Usage on Vehicles.** It also should only be performed after a rank vehicle function. It will assign vehicles, starting with the 1st ranked vehicle, to the driver that operates the vehicle most frequently.

Assign Vehs Using DA Override Fields

This assignment operation assigns drivers to vehicles based on the information passed in the field file. It requires three fields:

- The Vehicle field AssignmentOverride (field ID 400), which is used to signal that the vehicle's assignment can be overridden. Use a 1 to indicate that the assignment can be overridden or a 0 to indicate that the assignment cannot be overridden.
- The Driver field DriverID (field ID 367), which contains a driver number for the current driver.
- The Vehicle field AssignedDriver (field ID 17), which contains the number of the driver (based on DriverID) that should be assigned to the current vehicle.

Miscellaneous Functions

In driver assignment, there are a few functions that do not fit into a particular category. All miscellaneous functions are standalone, meaning they do not require a predefined algorithm.

Miscellaneous functions are only available from within a driver assignment scenario.

To add a miscellaneous function to your driver assignment scenario, select **Functions - Miscellaneous** from the list of **Available Variables, Functions, and Constants** drop down and then double-click the miscellaneous function you want to add.

Available Miscellaneous Functions

Clear Drv Ranking

When ranking drivers, the system allows for cascading ranking operations. This function breaks the cascading ranking that is inherent when using back-to-back ranking operations. This clear function should always be performed after a rank driver operation, when cascading is not desired.

Clear Drv Ranking (Usage Set)

When ranking drivers, the system allows for cascading ranking operations within a usage set. This function breaks the cascading ranking that is inherent when using back-to-back ranking operations. This clear function should always be performed after a Rank Driver (Usage Set) operation, when cascading is not desired.

Clear Veh Ranking

When ranking vehicles, the system allows for cascading ranking operations. This function breaks the cascading ranking that is inherent when using back-to-back ranking operations. This clear function should always be performed after a rank vehicle operation, when cascading is not desired.

CREATING A DRIVER ASSIGNMENT

Driver assignments can be created and added to standard programs and template generated programs. Local driver assignments will be available in the program where you created it and can be copied to other programs within the . Global driver assignments will be available across the .

Local and global driver assignments are created and function in much the same way. One difference is that global driver assignments can use only global table variables, fields and result variables. Local driver assignments can use both local and global elements. Another difference is that global driver assignments can be created from the globals tab and from the program details tab.

To Create a Driver Assignment

1. On the Program Details tab of the program where you want to create a new driver assignment, select either the Global Driver Assignment folder or the Local Driver Assignment folder.

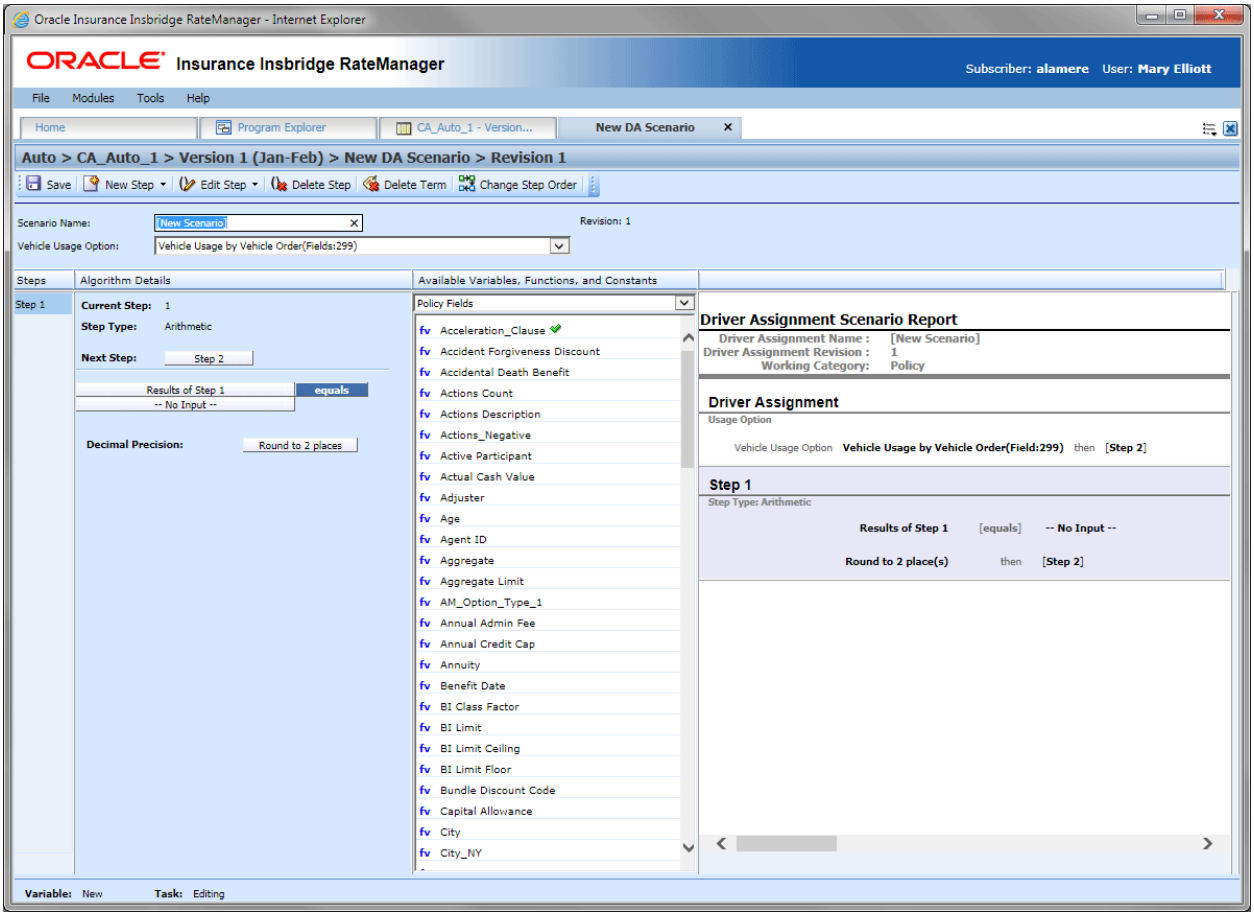


Figure 412 Creating a Driver Assignment

2. Click **New** in the top bar menu. You then need to select to create a driver assignment scenario or a driver assignment algorithm. If you do not want to select a folder, you can click New and select from there.

   New→Globals→Driver Assignment→DA Scenario

New→Globals→Driver Assignment →Flag Driver
New→Globals→Driver Assignment →Rank Driver
New→Globals→Driver Assignment →Flag Vehicle
New→Globals→Driver Assignment →Rank Vehicle
New→Locals→Driver Assignment →DA Scenario
New→Locals→Driver Assignment →Flag Driver
New→Locals→Driver Assignment →Rank Driver
New→Locals→Driver Assignment →Flag Vehicle
New→Locals→Driver Assignment →Rank Vehicle

3. This opens the Driver Assignment tab for the new driver assignment where you can enter or select the variable elements you need.

4. Enter a **Name** for the driver assignment. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

5. **FOR AN ALGORITHM:** Choose a **Working Category**. You can select the working category from the drop down menu or you can browse for a working category by clicking **BROWSE**. This will bring up the Browse Category window.

   From here you can navigate to the Category you want. Once you have selected the category, click OK. The category auto fills in the working category section. If this is not the screen you want, click **Cancel** to close the window and return to the previous screen.

NOTE: *Always save a driver assignment algorithm after defining the name, and working category. The driver assignment is not available to be used within a step until the variable has been saved.*

6. **FOR A SCENARIO:** Choose a **VEHICLE USAGE OPTION**.

7. The first step is populated in the Algorithm Details area. This must be a mathematical step. If you do not want an arithmetic step as your first step, add the desired step and then delete the arithmetic step. To do so, highlight **Step 1** and click **Delete Step** in the menu bar.

8. To add a new step to the driver assignment, click **New Step** and select the type of step you want to add. The screen refreshes with the new step parameters. Depending upon the step type, you can select the appropriate variables, functions, constants or fields. The values required are listed on the step.

   For example, if the step has an “Enter Variable” box, you must choose the variable type from the menu in the Available Variables, Functions and Constants area. The options are populated underneath. Click the “Enter Variable” box and then double-click the corresponding variable you want. The box refreshes with your selected value. For more information on the types of steps, see Working with Steps.

9. Continue entering steps until complete. When you are finished making your entries, click **Save**.

NOTE: *When selecting Callouts for use in a driver assignment or algorithm, the driver/vehicle category cannot be mapped.*

Adding a Variable, Result Variable or Custom Value

A new field variable, result variable or custom value can be created while entering or editing a driver assignment, without leaving the Edit screen.

- To add a field **Variable**: See Adding a New Field
- To add a **Custom Value**: See Custom Value
- To add a **Result Variable**: See Adding a Result Variable

EDITING A DRIVER ASSIGNMENT

Editing a driver assignment changes information about the scenario/algorithm such as name, vehicle usage option or working category and steps. Driver assignments that are locked cannot be edited. Template generated driver assignments that are using revision 1 are not open for editing. If you want to edit a template generated driver assignment, you must create a new revision. The new revision will be open for editing except for the name. The name cannot be changed. Changes are effective immediately. Global driver assignments can be edited from the globals tab or from the program details tab.

NOTE: *If a change is made to a global driver assignment, the change affects all programs that use the global driver assignment. If a change only needs to be made to one particular program, create a local driver assignment in the program and have the program reference the local driver assignment instead of the global.*

To Edit a Driver Assignment

1. On the Program Details tab, select the **Driver Assignment** you want to edit.

2. Double click or right click and select **Edit**. The Edit Driver Assignment tab opens with the details of the driver assignment.

3. Make any changes.

   To change the step order, you have two options:

   - **Option 1:** Select a step in the Steps column, drag it to a new location in the list and drop it.

   - **Option 2:** Select the step you want to move and click Change Step Order. A separate screen is displayed.
     - The current step position will be displayed at the top of the screen. Select where you want the step to be placed.
     - Click OK. You will be returned to the Edit Algorithm tab. Your step should be placed where you chose.

   To delete a step, select the step in the Steps column and then click Delete Step. You cannot delete the last step from an algorithm.

4. Click **Save** to save your work.

WARNING: Category Updates to Revisions

Category updates should not be done to a revision. All revisions of a variable should be at the same category level. Changing the category for only some of the revisions of a variable is not supported.

GLOBAL AND LOCAL COPIES

Driver Assignments may be copied in any program. Local driver assignments can be copied to the same program or to another program within the same product at the local level. Global driver assignments can be copied to the same program at the global level or, to the same program at the local level, or to another program within the same product at the local level.

If you are copying a global driver assignment, you will have the option to copy to the global level or to the local level. The availability of the driver assignment depends on where the variable is copied to:

- **Local variables** can be copied to the same program or to another program at the local level within the same product.
- **Global variables** can be copied:
  - At the global level and are available to every program in the product
  - To the local program level
  - To another program within the same product at the local level

The copy options depend on whether the driver assignment to be copied is at the global level or the local level.

When copying a driver assignment, the structure (including the vehicle usage options/category and steps) is copied. Only the active revision is copied. If you have a driver assignment with 10 revisions, and you copy the driver assignment, only the active revision is copied into the new drivers assignment. The other 9 revisions are not copied.

Global Copies

A global copy creates a copy of the driver assignment and places it at the global level. There are no options.

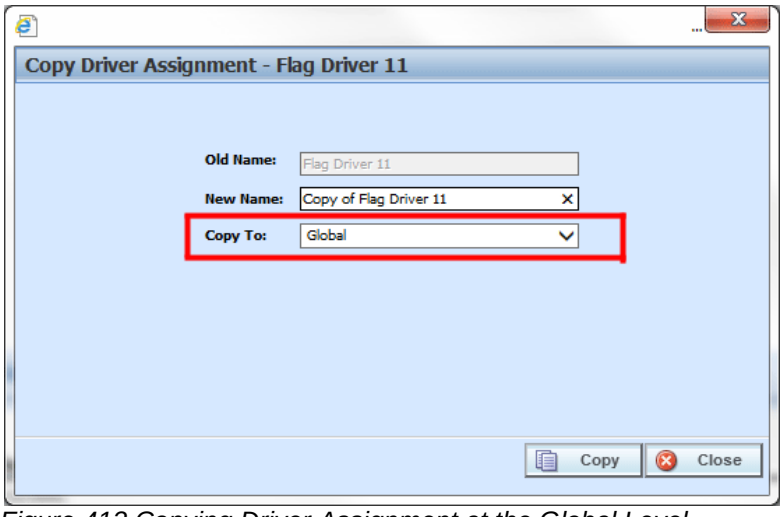


Figure 413 Copying Driver Assignment at the Global Level

When copying a global variable to a local level:

- Global copies are done at the product level. The Destination Program name is listed first and the folder name is in parenthesis.
- The New Name must be different from the Old Name.

Local Copies

When a global driver assignment is copied to a local level or when a local driver assignment is copied, there are two copy options:

- **Existing Program:** Copy the driver assignment into the existing program under another name.
- **Other Program:** Copy the driver assignment into another program within the same product.

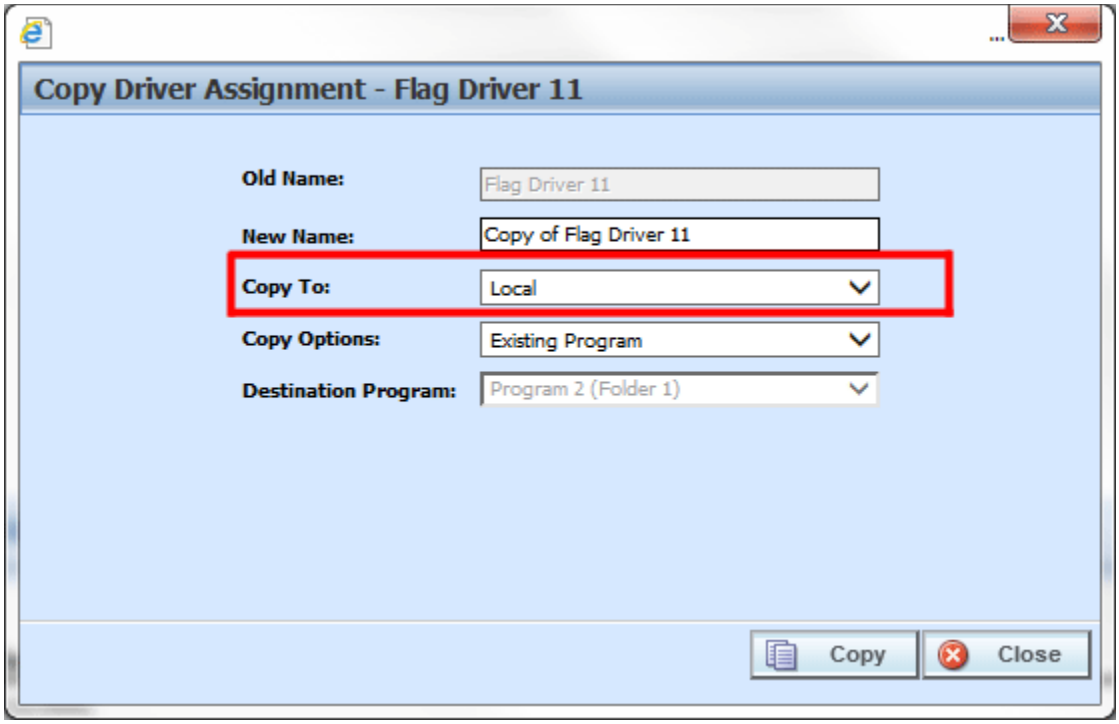


Figure 414 Copying Driver Assignment at the Local Level

Local copies can be done to the local level of any program in the product. When copying a variable to a local level in another program:

- Local copies are done at the product level. The Destination Program name is listed first and the folder is in parenthesis.
- Local to local copies check for dependencies when the copy is made:
  - When performing a local to local copy, dependencies such as the usage of other local elements are checked. If there are local elements used in the element being copied that are not in the Destination Program, then those elements are copied also.

  - For example, the Option AB Fee algorithm, a local variable, uses variables Option A and Option B, also local variables. When Option AB Fee is copied from program Maine to program Kansas, the algorithm is copied and so is the variables that are being used.

- Local to local copies do not require a name change:
  - When copying to the local level of a different program, the variable name cannot be changed. The New Name is grayed out. This creates an identical variable in the other program and reduces the need for editing. If a change is made to the original variable and then the variable is copied for a second time, the copy is created as a new revision in the Destination Program.
  - When copying at the local level in the same program, the name must be changed.
- Copies made in the current program must have a new name.

COPYING A DRIVER ASSIGNMENT

Global driver assignments can be copied from the globals tab or from the program details tab. Unlike a local driver assignment, you will not be able to select copy options.

1. Select the driver assignment you would like to copy and right click. Select **Copy** from the menu.

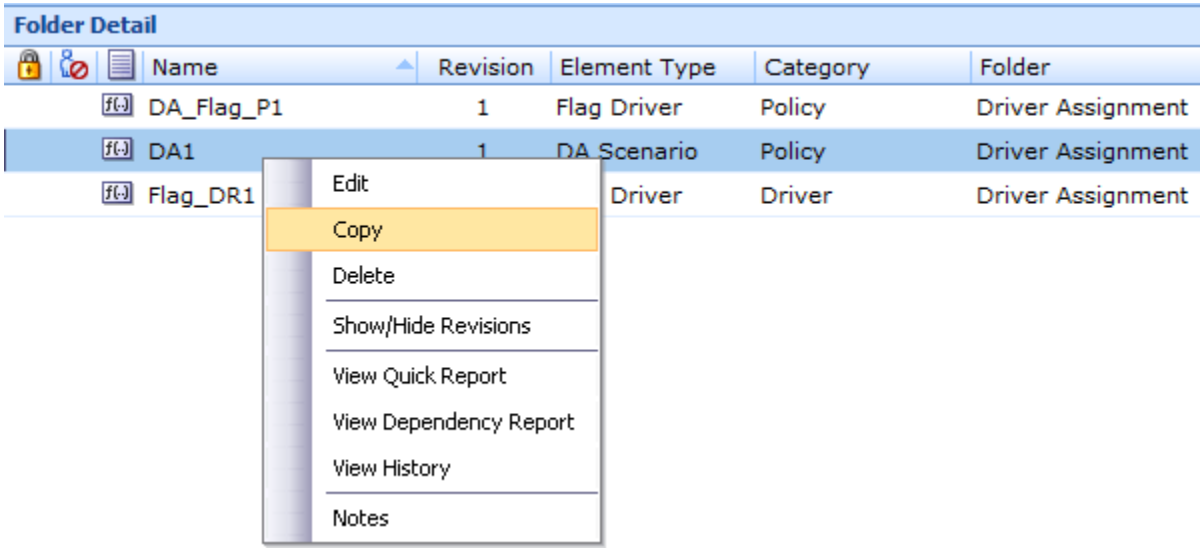


Figure 415 Copy Driver Assignment Menu

2. A separate window is displayed.

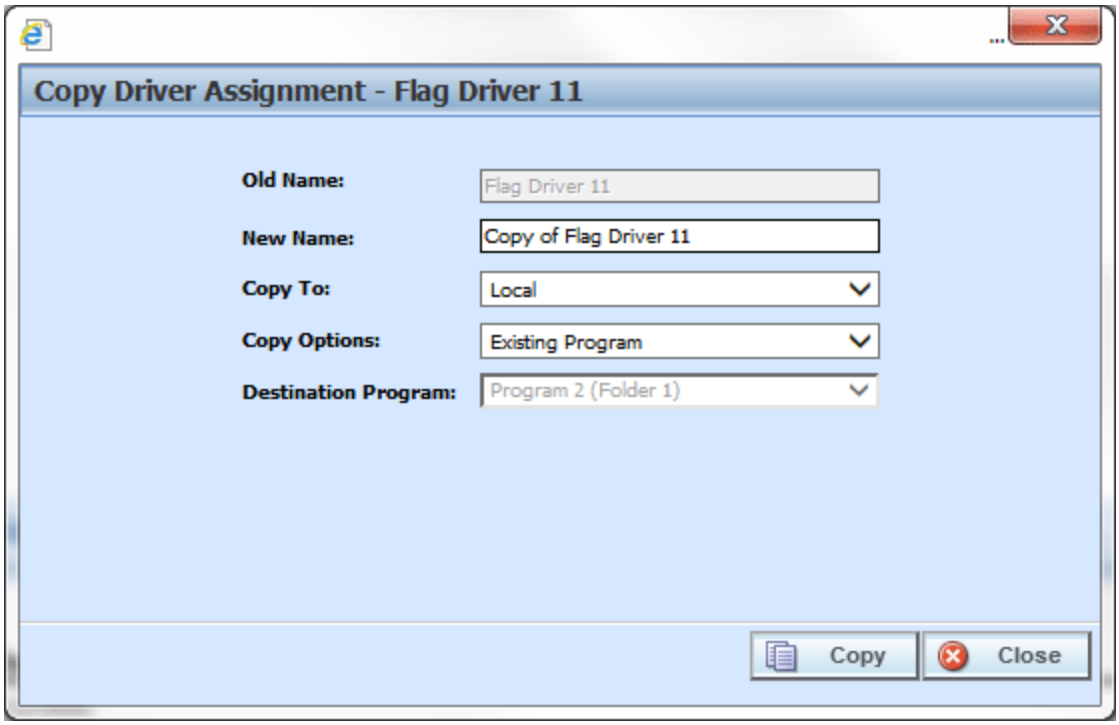


Figure 416 Copy Driver Assignment

3. Select the **Copy Options**. It is recommended that you change the copy options first. Changing options may reset the name field.

4. If this is a global driver assignment, select if this is a local or global copy. This option is not present on local driver assignment.

5. The copy options determine the Destination Program selections. If you are copying to the current program, you cannot choose the destination program. If you are copying to another program, select the **Destination Program** from the drop down menu.

6. Enter in a **New Name**. The name can be entered first. It is recommended that you change the copy options first. Changing options may reset the name field.

7. Click **Copy** to copy the driver assignment. Dependencies will be searched. If there are any dependencies that need to be resolved, a window is displayed. See Resolving Dependencies.

8. If the copy was successful, you will be returned to the previous screen. Your driver assignment will have been copied. The last updated date will be the date the copy was made.

DELETING A DRIVER ASSIGNMENT

When a driver assignment scenario or algorithm is no longer needed, it can either be left in RateManager or deleted. When you delete a driver assignment scenario or algorithm:

- The driver assignment scenario or algorithm gets deleted from all versions of the program.
- All revisions of that driver assignment scenario or algorithm get deleted as well.

- When you delete a driver assignment algorithm, it will be deleted from any driver assignment scenario that is using it.

If you are uncertain as to whether the driver assignment will be used again, leave it in RateManager. This will save time if the driver assignment is needed again and leaving it in RateManager will not slow down rating. If you try to delete a driver assignment scenario that is being used in a sequence, you will receive an error. Driver assignments that are locked or that are template generated cannot be deleted.

NOTE: *Driver Assignment scenarios that are being used in a sequence cannot be deleted.*

Any unlocked global driver assignment that is not in use can be deleted. Global driver assignments that are in use, locked, or template generated cannot be deleted. If you try to delete a global driver assignment that is unavailable for deletion, an error message will be displayed informing you where the global driver assignment is being used.

A lock icon indicates that this driver assignment is locked by either a program or a template and is not open for editing or deleting.

Global driver assignment revisions can be deleted from the globals tab or from the program details tab.

To Delete a Driver Assignment

1. Navigate to the **Program Details tab** for the program that contains the driver assignment you want to delete.

2. Select the driver assignment you want to delete and right click. Select **Delete** from the menu.

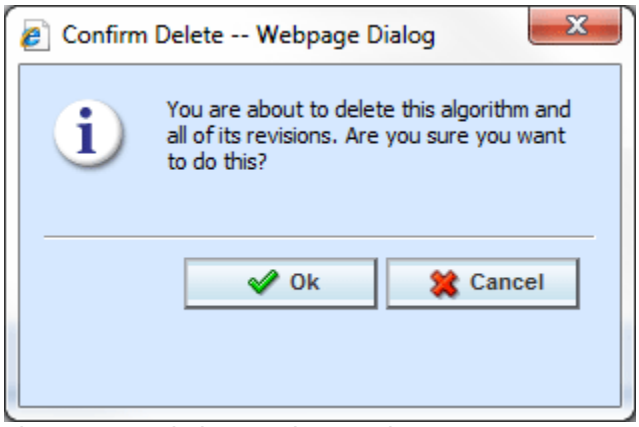


Figure 417 Deleting a Driver Assignment

3. You will be asked to confirm the deletion. Select **Ok** to delete the driver assignment and refresh the list of driver assignment. Select **Cancel** to return to the Program Details tab.

NOTE: *If you try to delete a driver assignment scenario that is being used in a sequence by any program version, you will receive an error message.*

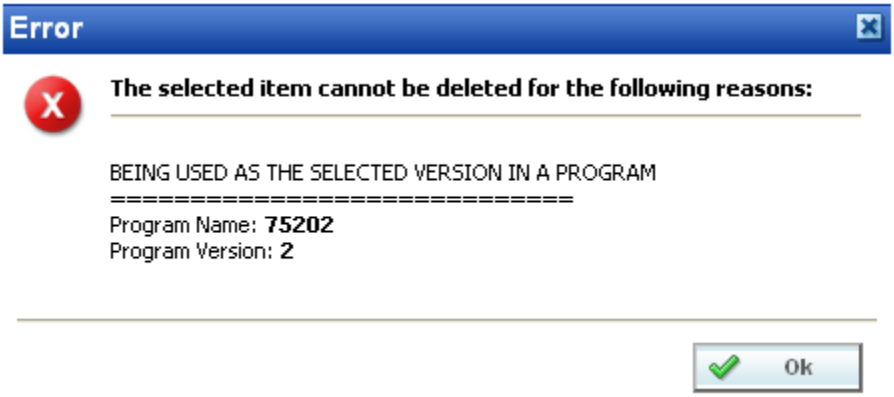


Figure 418 Driver Assignment Delete Error Message

CONSTANTS

In RateManager, a constant is a fixed value that can be used in calculated and table variables, algorithms, driver assignments, and filter rules. There are two types of constants in RateManager, pre-defined and user defined. Pre-defined constants are located at the system level. These are system defined values that are available to every program regardless of domain type or project. User defined constants can be created at the system level, the universal level, or the global level and can be edited, and deleted at any time.

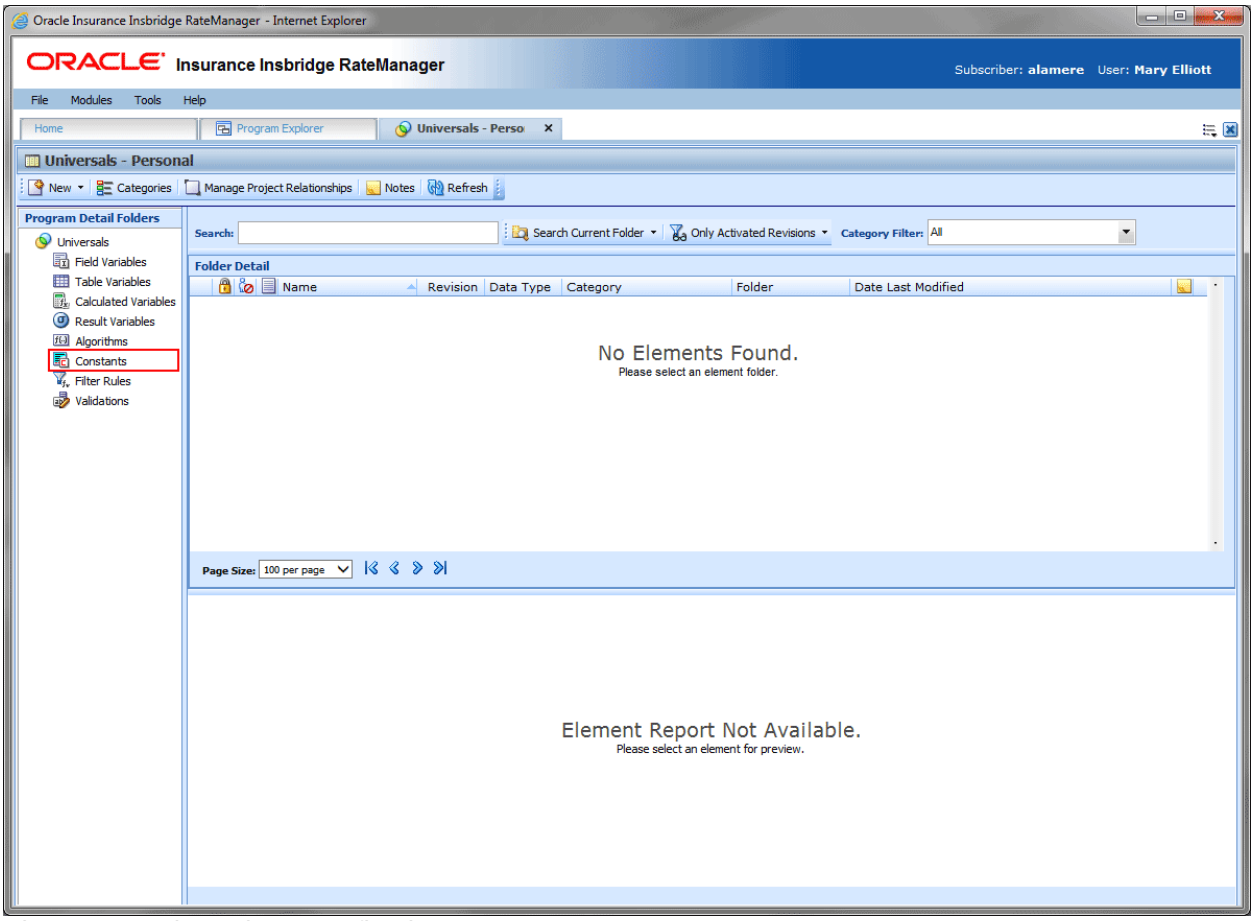


Figure 419 Universal User Defined Constants

Constant Type Details

There are two types of constants in RateManager:

- **Pre-defined:** Pre-defined constants are fixed values and system values that are available to every program in the Insbridge system. This type of constant is built in to the Insbridge system and can be viewed in Tools→Setup→System Level Constants or from the System Level Constants icon on the Home tab. Pre-defined constants have two types:

- **Fixed:** These constants have a fixed value. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 1000, PI, 10000, Y, YES, N, NO, and NULL. The value of these constants cannot be changed. If you want a different value, you can use a Custom Value, create a field with a specific value, or create a user-defined constant with a specific value.
- **System Value:** A constant value that comes from the system. System values can also be used as table variable criteria. There are five system values:
  - **Current Category Instance #**: This is the instance that the category is currently on. For example, you may have an auto program where you can enter multiple vehicles. Each vehicle would have its own instance in the vehicle category. If you limit personal auto to 9 vehicles, you may want to set a rule or trigger another action if the vehicle instance is 10.
  - **Current System Date and Time**: Current server time and date at the time of execution.
  - **Current System Date**: Current server date at the time of execution.
  - **Current Program Version**: The current program version being used at the time of execution.
  - **Current Rating Environment**: The current rating environment being used at the time of execution.

NOTE: *Pre-defined constants are managed in another area, Tools→Setup→System Level Constants. When working on constant, make sure you are in the correct location.*

- **User defined:** User-defined constants, also called domain constants, are values that are created by system users and consist of three types:
  - **System Level:** These are found in every project regardless of domain type, or domain and can be viewed in Tools→Setup→System Level Constants or from the System, Level Constants icon on the Home tab.
  - **Universal Level:** Universal constant are created and managed at the universal level and are project specific. Universal constants can be managed only from the Universal tab. Universal constants can be viewed on the Program Details tab but no edits, copies, or deletions can be done. Every program in the project has access to universal level constants.
  - **Global Level:** Global constants are created and managed at the global level and are product specific. Global constants can be accessed from the Global tab or from the Program Details tab. Every program in the product has access to global level constants.

Constant Listing

Constants can be used in table variables, calculated variables, algorithms, drivers assignments, and filter rules. In the Available Variables, Functions and Constants list, system level constants are listed with pre-defined constants first, followed by user-defined constants and custom values last. User defined domain constants are in a separate list.

Navigating to Constants

Constants are not an option on the Program Explorer tab.

There are three paths to view or manage constants, either from the universals tab, globals tab or from the program details tab. The universal tab displays universal constants only. The global tab displays global constants only. You cannot view or manage universal constants on the global tab or view or manage global constants on the universal tab. Constants are not created at the local level. The views and options are the same on either tab.

1. Modules→Program Explorer to open the Program Explorer tab.

2. Select the domain type→domain→project→product→folder→program→program version. You also can access Constants from the globals tab. Domain type→domain→project→product→ globals or the universals tab via domain type→domain→project→universals.

3. **Open** the program version to get the Program Details tab. Constants are available from the Program Detail Folders area on the left side of the tab.

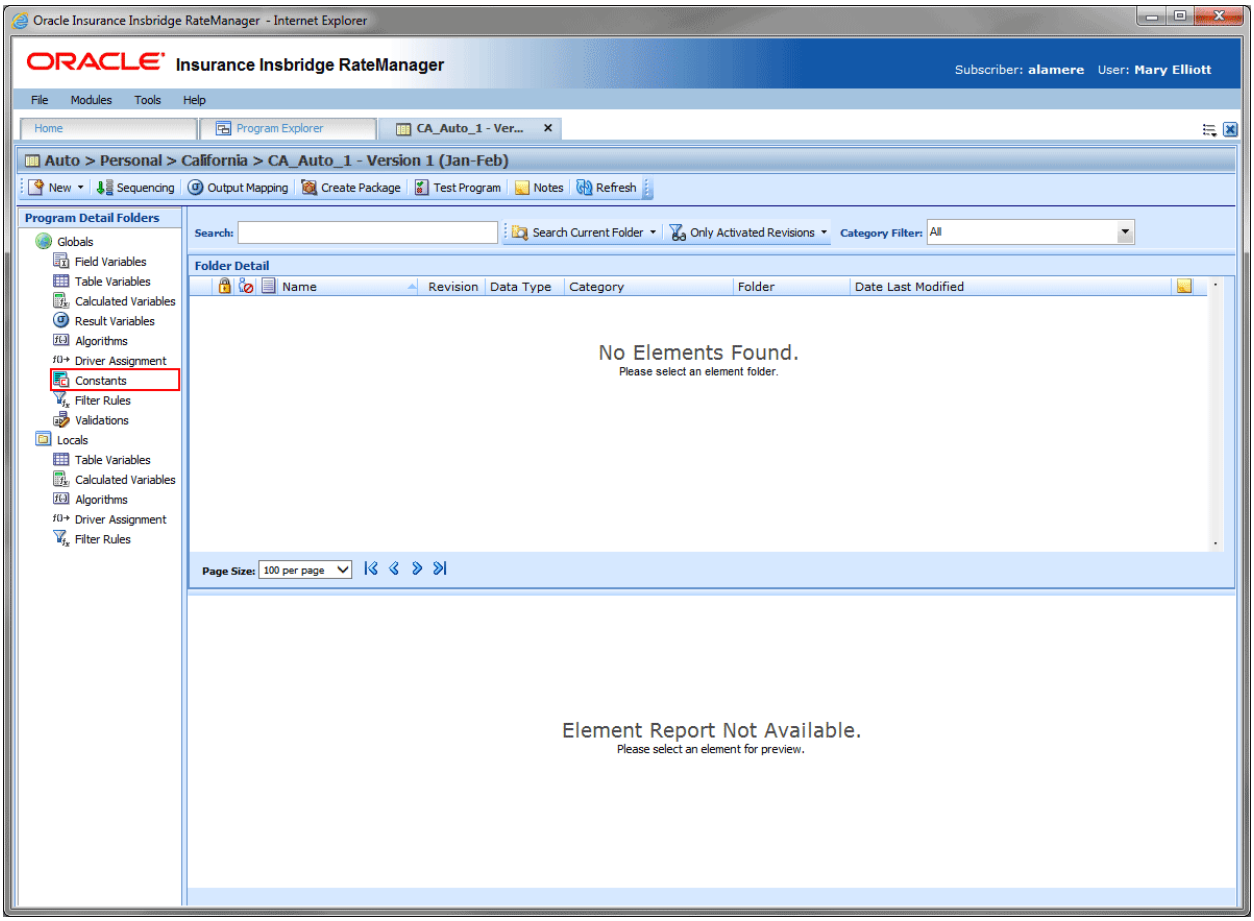


Figure 420 Constants Listing on Program Details

CONSTANTS FUNCTIONALITY

Functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any constant. The functions you can perform depend on the status of the constants. Locked constants have fewer options. There are icons that can tell you at a glance what the status of the constants is.

Icon	What it Indicates
	Constants. Constants can be created on the universal tab, the global tab, or from a program version. The same options, new, edit, copy and delete are available on each tab.
	Locked Constants. Locked constants are not open for editing. A lock icon indicates that this constant is locked by either a program or is system generated. System constants are not open for copying, editing, or deleting. Program locked constants can be copied. If you have to edit a program locked constants, you must return to the program and unlock it. This unlocks the elements and changes can be made.
	Template Lock. If the constant is locked because it came from a template, you cannot unlock the constants. You can copy the constants or create a new constant for use in the program if necessary.
	In-Use. Constants that are in use are currently being worked on by another user in the system are not open for editing, copying, or deleting. The constants will be available when the user exits the constant.

CONSTANTS SCREEN DETAILS

The screen details are the same for the Universal tab, Global tab and program details. For more on screen detail descriptions, please see Program Details→Screen Details.

Right Click Menu Options

To view right click menu options, highlight the constants you want to view and right click. Unlocked constants variables have six right click menu options:

- **Edit:** Edits the constants.
- **Copy:** Brings up the copy screen for the constants.
- **Delete:** Removes the constants from the list. A warning message is displayed to confirm the deletion. You must have permissions to delete.
- **View Where Used:** Displays an information screen that details where the variable is being used. See View Where Used for more information.
- **View History:** Displayed the Audit History report. See View History for more information.
- **Notes:** Enter, edit, and view notes for the constants.

Template locked constants and program locked constants have four options, Copy Variable, View Where Used, View History, and Notes.

CREATING A CONSTANT

Template generated constants are locked and not open for editing, or deletion. A user created constants can be added to a template generated program if needed. The new constants are open for editing, copying, or deletion.

1. On the Program Details tab, select Constants from the Program Detail Folders area. Click New in the top bar menu. This opens the New Domain Constant window. For a universal constant, select Constants from the Program Detail Folders area on the universal tab. Click New in the top bar menu.

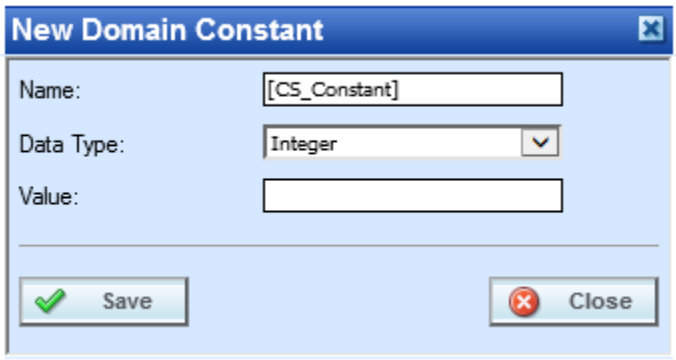


Figure 421 Universal Constants Variable

2. Enter in a Name. Names can be up to 40 alpha numeric characters and allowed special characters. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Select a Data Type from the menu. Data type cannot be edited once saved.

4. Enter the Value. The value you can enter depends on the data type.

   - Integers – Values can be up to 21 numeric digits.
   - String – Values can be up to 50 alpha numeric characters and allowed special characters.
   - Decimal – Values can be up to 50 numeric digits with a single decimal point.
   - Date – Values can be up to 50 alpha numeric characters and allowed special characters.

5. Click Save to save your constants. The screen refreshes and the new constant is displayed on the Program Details→Constants tab.

EDITING A CONSTANT

Unlocked constants can be edited at any time. If the constant is locked because the program is locked, you must return to the program and unlock it. Once unlocked, changes can be made to the Name and value. The data type cannot be edited. If the constant is locked because it came from a template, you cannot unlock it.

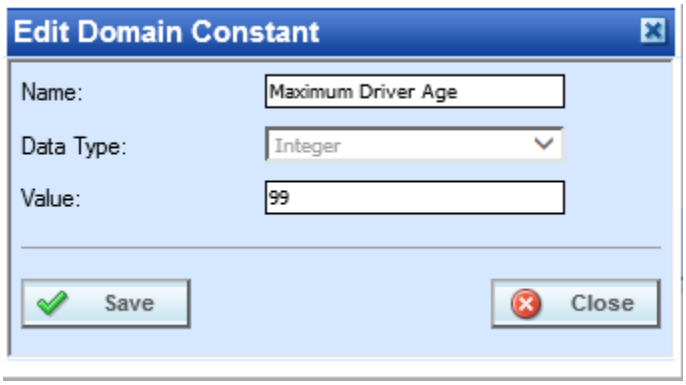


Figure 422 Editing a Universal Level Constant

NOTE: *Please take care when editing or deleting user-defined constants. When values are updated, the updates will affect every program where that constant is being used. The update happens at the time of packaging. This action may cause unintended changes in other programs. To see where the user-defined constant is being used, run a Where Used report.*

Data type cannot be edited. If you have made a mistake with the data type, you will have to create a new constant with the correct data type.

1. Either double click the constant you want to edit or right click and select Edit. This opens the Edit Domain Constant window.

2. Make any changes to the Name and Value. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Click Save to save your changes.

COPYING A CONSTANT

Constants can be copied at any time. The constant is copied with the exact same data type. To make a change to the constant name or value, you must go into the edit screen. Names cannot be duplicated within the domain. Data types cannot be edited. If you need a different data type, you will have to create a new constant. .

1. Right click the constant you want to copy. Select Copy. The Copy Constants window is shown.

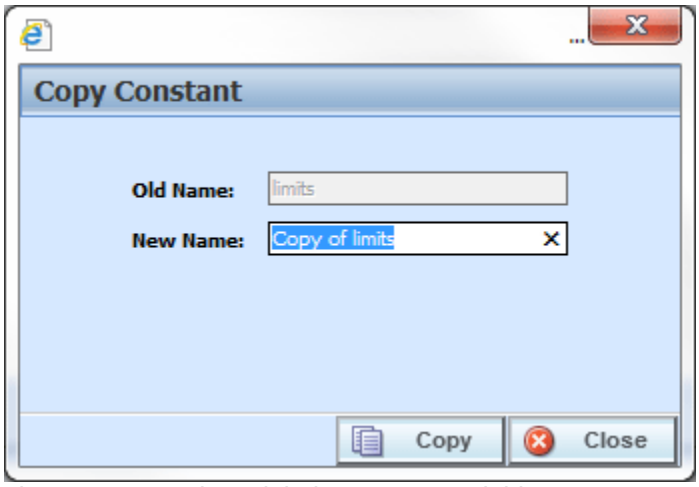


Figure 423 Copying Global Constant Variable

2. Enter in a New Name. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

3. Click Copy to copy your constants. The screen refreshes and the copied constant is shown on the Program Details→Constants tab.

DELETING A CONSTANT

Any unlocked constant that is not in use can be deleted. Constants that are in use, locked, or system generated cannot be deleted. If you try to delete a constant that is unavailable for deletion, an error message is displayed informing you where the constants are being used.

NOTE: *Take care when deleting Constants, this action cannot be undone.*

1. Constants can be deleted by right clicking the constant and selecting Delete.

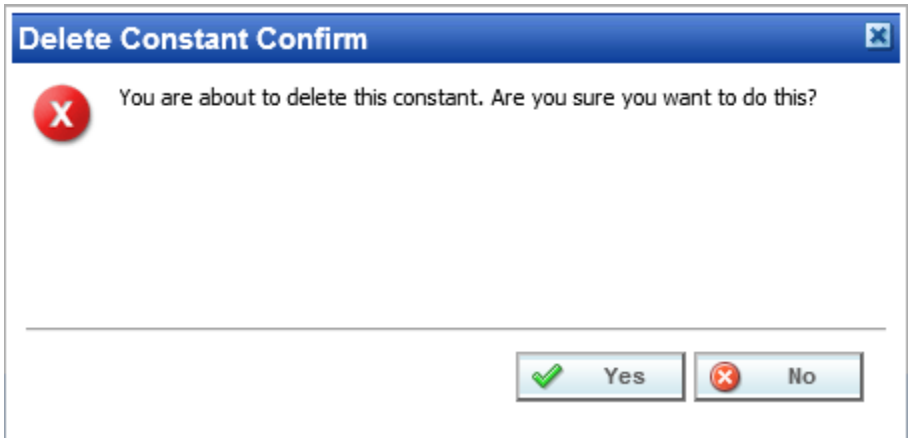


Figure 424 Deleting a Constant

2. A confirmation message is displayed.

3. Click OK to delete or Cancel to cancel. If the constant is being used, you cannot delete it. A message is displayed informing you where the constant is being used.

Chapter 22

FILTER RULES

Filter rules are assigned to table variables, calculated variables, or algorithms to determine if the variable should be run. When a filter rule is applied, for example to a table variable, then the table variable will only be executed when the rule is met.

When a filter rule is run, it returns either a true or false value. This value determines if the variable or algorithm needs to be run. For example, if the value is true then the variable or algorithm is run and if the value is false, the variable or algorithm is not run. Filter Rules are an easier way to run table or calculated variables in certain conditions.

A calculated variable, a table variable, or an algorithm can have one filter rule applied. This rule is run first. The value returned determines if the variable or algorithm needs to be run. Only one rule can be assigned.

Filter rules and variables have to be on the same category level. Filter rules can have multiple steps and consist of constants, step variables or fields. No other element can be used to create a filter rule. Regardless of how many steps are used, the last step in the filter rule must specify if the rule should return a true or false value. Naming standards can be applied. Filter rules can be assigned to a calculated variable, table variable or algorithm only and they cannot be used as criteria.

Filter rules are in reports. The debug report has a return true or false in step banner where it is used.

Filter rules can be created at three levels:

Universal: Universal filter rules are created and managed at the universal level and are project specific. Universal variables can be managed only from the Universal tab. Universal variables can be viewed on the Program Details tab but no edits, copies, or deletions can be done.

Global: Global variables are created and managed at the global level and are product specific. Global variables can be accessed from the Global tab or from the Program Details tab.

Local: Local variables are created and managed at the program version level and are specific to the program. Local variables can be accessed from the Program Details tab.

There are a few differences between universals, globals and locals. One difference between them is that universal filter rules are available to all programs in the project. Global filter rules are available to all programs within the product while program level filter rules are available only to the program where they were created. Another is that universal filter rules can use only universal elements. Global filter rules can use universal and global elements. Local filter rules can use universal, global, or local elements.

Filter rules can be template generated or created by you. Filter Rules that are user created are open for editing, copying, creating new revisions and deletion if they are not locked.

Template generated filter rules are locked and cannot be unlocked. No editing or deleting is allowed but they can be copied. This is to ensure the integrity of the template. New filter rules can be added to a template and new revisions can be created. These new filter rules are open for editing, copying or deletion, unless the program version is locked.

FILTER RULES OPTIONS

- Creating a Filter Rule
- Editing a Filter Rule
- Copying a Filter Rule
- Deleting a Filter Rule
- Creating a Filter Rule Revision
- Changing the Active Revision of a Filter Rule
- Deleting a Filter Rule Revision
- Creating a Custom Value
- Resolving Dependencies

Navigating to Filter Rules

There are three paths to view and manage filter rules; Form the Universals tab, from the Globals tab or from the Program Details tab. The universal tab displays the universal filter rules only. The Global tab displays global variables only. You cannot view or manage local variables on the Global tab or the Universals tab. On the Program Details tab, you can select to view either global filter rules or local filter rules. If you want to view and manage local filter rules, you must use the Program Details tab. The views and options are the same on either tab.

From either tab, when you select Filter Rules in the Program Details Folder, the current filter rules are listed in the Folder Details area.

Filter rules are not an option on the Program Explorer tab.

1. Modules→Program Explorer to open the Program Explorer tab.

2. Select the domain type→domain→project→product→folder→program→program version. You also can access calculated variables from the globals tab. Domain type→domain→project→product→globals. Universal filter rules are available from domain type→domain→project→universals.

3. Open the program version to get the Program Details tab. Filter rules are available from the program details folders area on the left side of the tab.

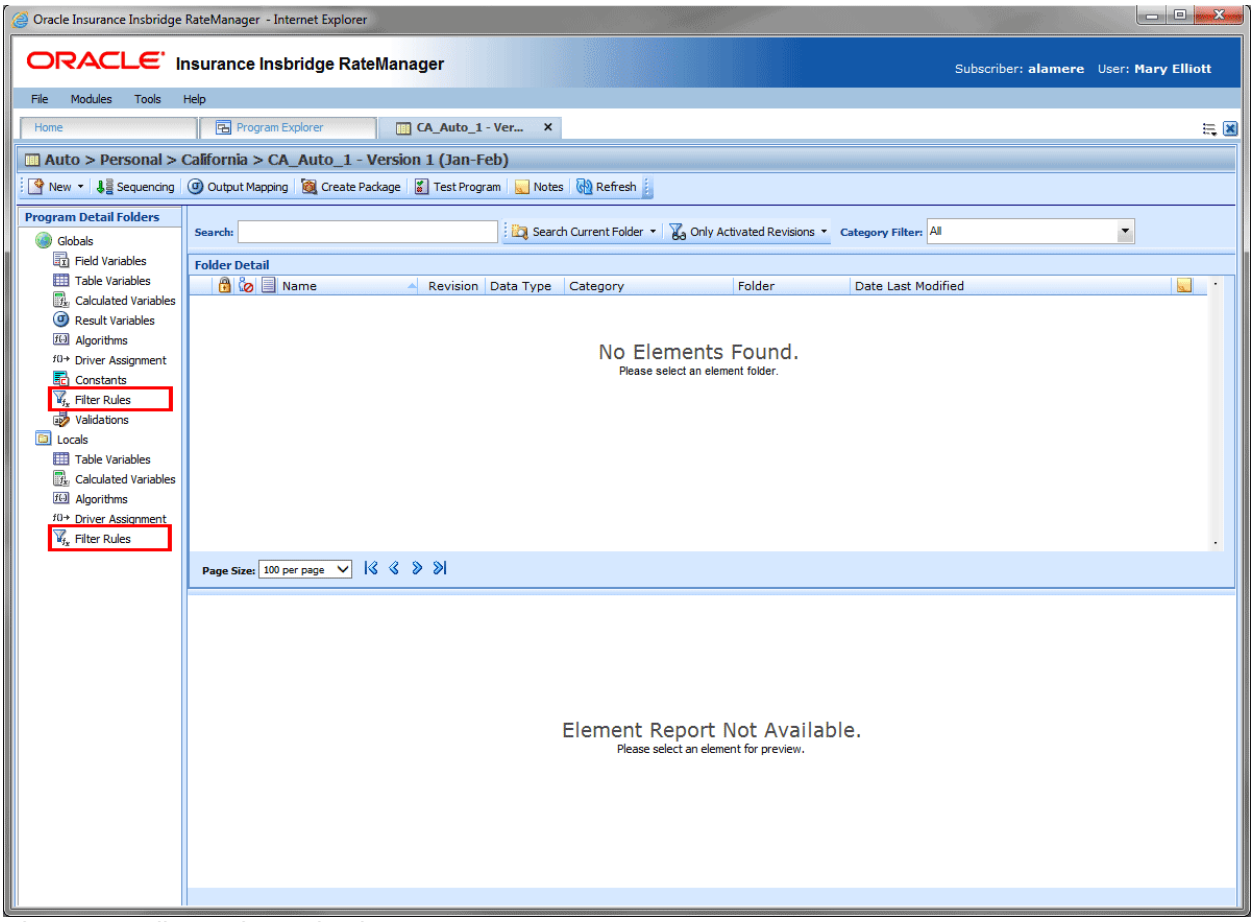


Figure 425 Filter Rule Navigation

FILTER RULE FUNCTIONALITY

Available functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any unlocked filter rule.

- **Filter Rule Functions:** The copy and delete functions can be performed on the Program Details tab using a right click menu. Reports also can be generated using the right click menu on the Program Details tab.
- **Filter Rule Revision Functions:** You can view, copy, and delete filter rule revisions. Active revisions can be set on the Program Details tab as well.
- **Filter Rule Tab:** When you create a new filter rule, edit an existing filter rule, or work with revisions, you will be placed on the Filter Rule tab.

The functions you can perform depend on the status of the filter rule. Locked filter rules will have fewer options. There are icons that can tell you at a glance what the status of the filter rule is.

Icon	What it Indicates

	Filter Rules. Universal filter rules are project specific and can be created on the Universal tab only. Global Filter Rules are product specific and can be created at the global level, or from a program version. Local Filter Rules are program specific and can be created at the program version level. The same options, new, edit, copy and delete are available on either tab.
	Universal. Indicates that the filter rule was created at the universal level. The filter rule is grayed out and can be edited, copied or deleted from the universal tab only.
	Locked Filter Rule. Locked Filter Rules are not open for editing. A lock icon indicates that this filter rule is locked by a program and is not open for copying, editing or deleting. If the filter rule is locked because the program is locked, you must return to the program and unlock it. This will unlock the filter rule and changes can be made.
	In-Use. Filter Rules that are in use are currently being worked on by another user in the system or variables that you have open in another tab are not open for editing, copying, or deleting. The filter rule will be available when the user exits.
	Active Revision. This variable is the current active revision.

Filter Rule Screen Details

The screen details are the same for the Universal tab, Global tab and Program Details. For more on screen details descriptions please see Program Details→Screen Details.

FILTER RULE TAB

The Filter Rules tab is available when you create a new filter rule, edit an existing filter rule, or work with revisions.

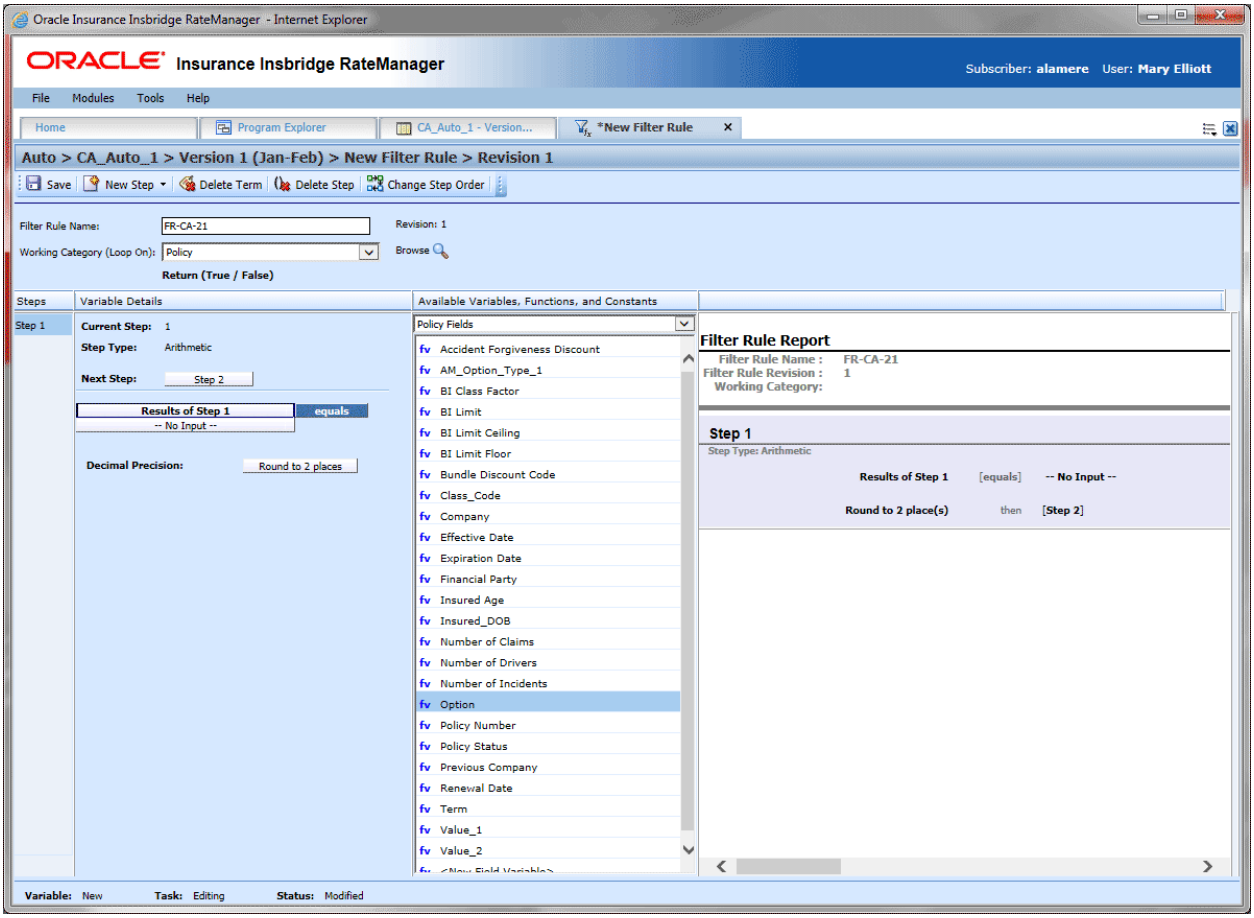


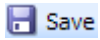
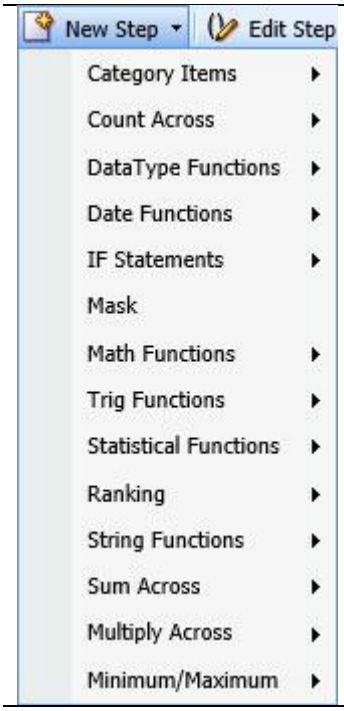
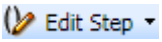
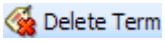
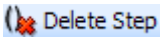
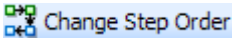
Figure 426 Filter Rule Tab

Filter Rule Rules

- The first step of a new filter rule is always an arithmetic step. If you do not want an arithmetic step as your first step, add the step type you do want as Step 2. You can then delete the automatic arithmetic step.
- You cannot have a filter rule without any steps. There must always be at least one step. You cannot delete the last remaining step.
- Regardless of how many steps are used, the last step in the filter rule must specify if the rule should return a true or false value.

FILTER RULE FUNCTIONALITY

The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform on this tab.

	**Save:** Saves the filter rules.
	**New Step:** Creates a new step in the filter rule. For more information, see Working with Steps.
	**Edit Step:** Copies the selected step or pastes the steps on the clipboard in the selected step order.
	**Delete Term:** Deletes the currently selected term from a step in the calculated variable.
	**Delete Step:** Deletes the currently selected step. You cannot delete the last step of a filter rule.
	**Change Step Order:** Allows users to change the order of the steps without using the drop and drag feature.

FILTER RULE SCREEN DETAILS

There are four areas on the Filter Rules Tab.

- **Filter Rule Naming Information:** The filter rule information area contains the name and defines the working category of the filter rule.
- **Filter Rules Details:** The details area is where you enter the steps that the filter rule will perform.

- **Available Variables, Functions and Constants:** The list of the available variables, step types and constants that can be used in the Filter Rules Details area.
- **Filter Rules Report:** A listing of all the steps currently in the filter rule.

Filter Rule Naming Information

The filter rule information area is where you define the filter rule.

Naming standards can be applied to system elements. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the element and for case enforcement to be applied to any newly created element or any existing elements that are being edited. If a naming standard is in place, the pattern is displayed in the element name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented for all elements.

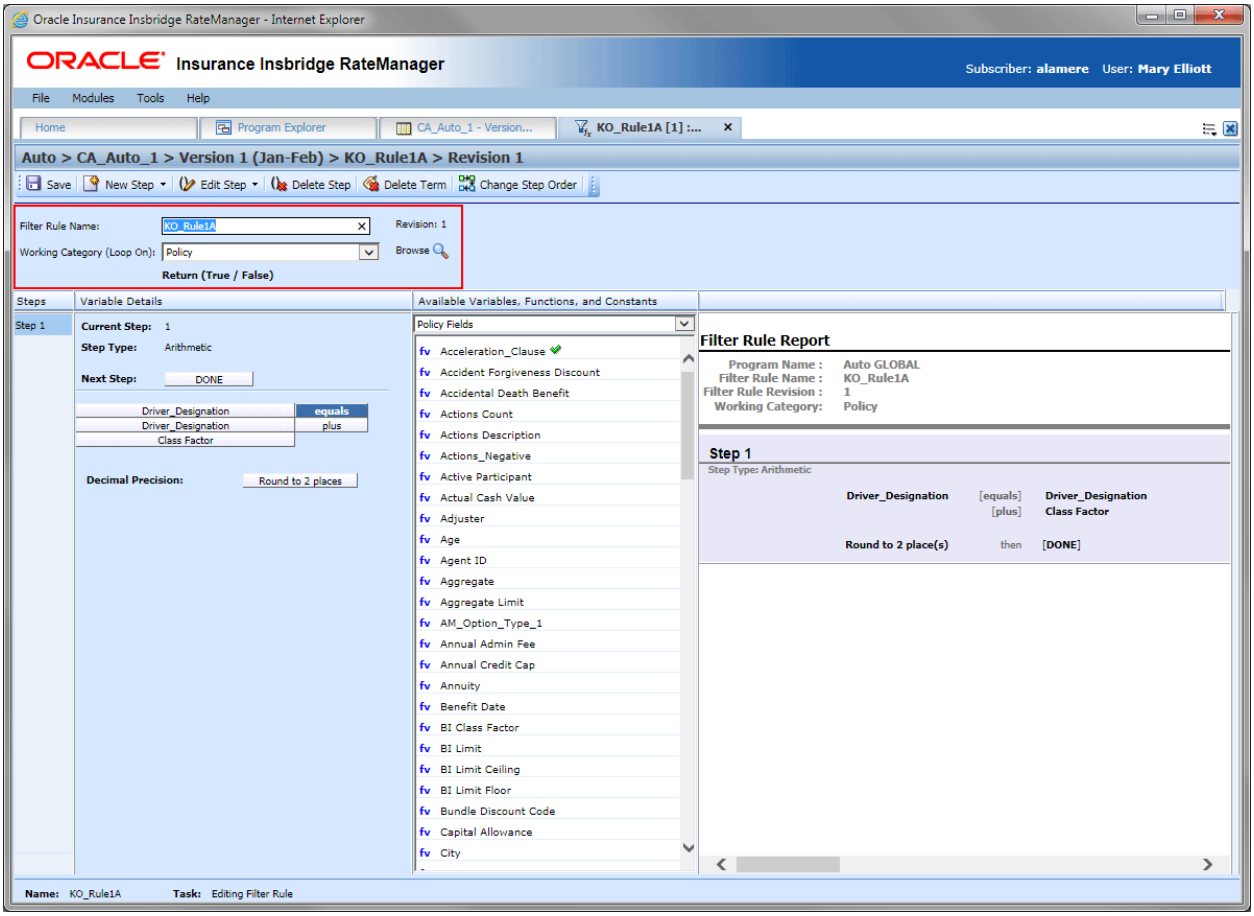


Figure 427 Filter Rules Naming Information

Screen Details

Filter Rule Name: Entry for what you want the filter rule to be called. Alphanumeric characters are permitted with limited special characters (ex: &, *, '', +, @, etc.). Please see

Restrictions for more information. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

Working Category: Sets the category where this filter rule will be run. For example, if you have an auto test case with three vehicles, and a filter rule to only run a table search when the vehicle is less than three years old, the vehicle category may be run three times or just once depending on the age of the vehicle.

You can select the working category from the drop down menu or you can browse for a working category by clicking **BROWSE**. This brings up the Browse Category window. From here you can navigate to the Category you want. Once you have selected the category, click OK. The category auto fills in the working category section.

The Working Category determines the options displayed in the Available Variables area. For example, if you select Policy as the Working Category, then the menu choices in the Available Variables area will be Constants, Step Variables, and Policy Fields. If you select Vehicle, the options will now include Vehicle Fields along with Constants, Step Variables, Policy Fields, Policy Variables and Policy Results.

Filter Rule Details

This area contains the steps that the filter rules will perform.

A new filter rule always begins with an arithmetic step as **Step 1**. If you do not want an arithmetic step as your first step, add the desired step and then delete the arithmetic step. To do so, highlight **Step 1** and click **Delete Step** in the menu bar at the top of the tab.

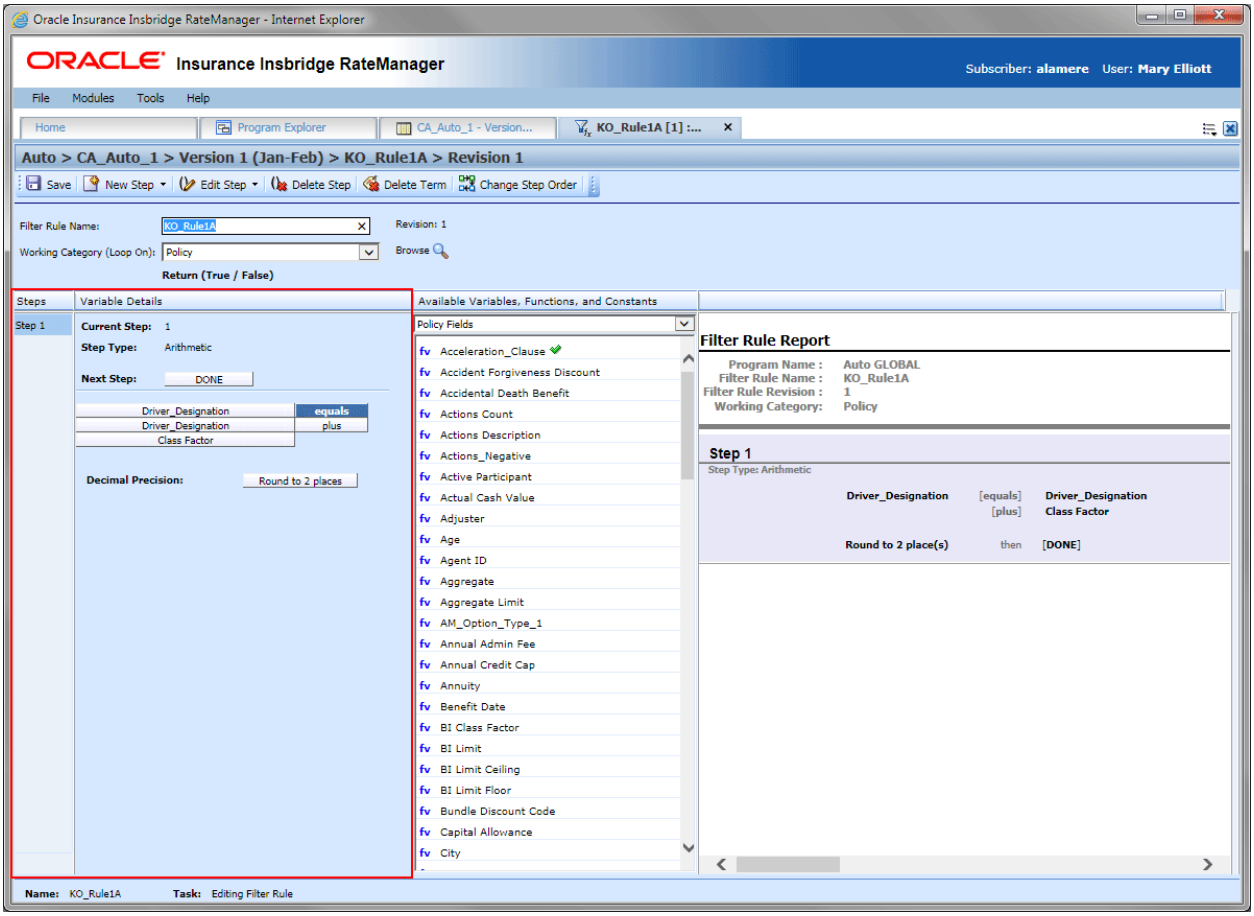


Figure 428 Filter Rules Variable Details

Screen Details

Steps: Sequential order of steps created with drag-and-drop capabilities for moving a selected step up or down in the sequence. The last step must define the return value, either true or false.

Variable Details: Shows the selected step number, step type selected, next step to follow and fields/variables/constants used by the rule. Rounding is set by clicking on the text box next to **Decimal Precision**. You can select a rounding precision from 0 to 10 digits.

Fields, step variables and constants each have a right click menu available; View Report, View Where Used, Custom Value, and Delete. Not all options are on every field, step variable, or constant.

Filter Rule Available Variables, Functions and Constants

The Available Variables, Functions and Constants area contains a listing of available fields sorted by category; system and domain constants; and step variables that can be used as criteria.

For Fields, you can add a new field variable without leaving the Edit Filter Rules tab. Scroll to the bottom of the available variables list and select <New Field Variable>. A window is displayed that allows you to enter a new field variable.

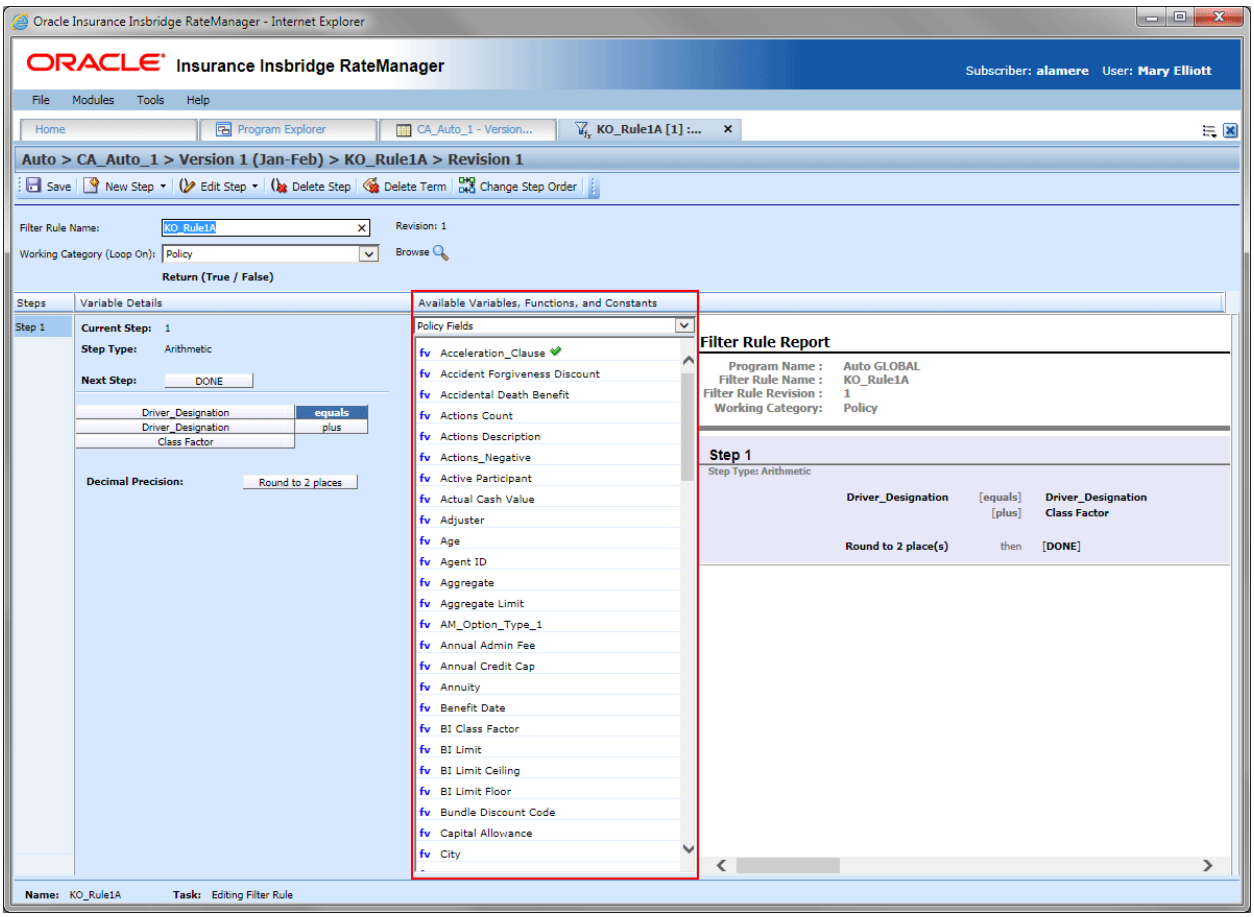


Figure 429 Available Variables, Functions and Constants

Screen Details

Menu: The menu is the top of the area. Click the down arrow to select the category you want. Policy fields are available for all categories. Driver and Vehicle fields are available in the Driver-Vehicle category. Categories are determined by the working category in the Variable Information area.

Selection: The available fields, constants and step variables are listed. Fields are based on the working category you select. Policy level fields are always listed. If you select another category, that categories fields will be listed.

Selection	Description
System Level Constants	Pre-defined constants, including category instances and system date and time, and system wide user-defined constants. Or you can create a custom value for use in this program.
Domain Constants	User-defined constants found only in a selected product.
Step Variables	Results from the previously entered steps.
Policy Fields	Fields in the Policy category.

Listing may also include other category fields.

Filter Rule Report

This area contains a detailed list of the steps you have entered. This information is view only. No actions can be performed in this area.

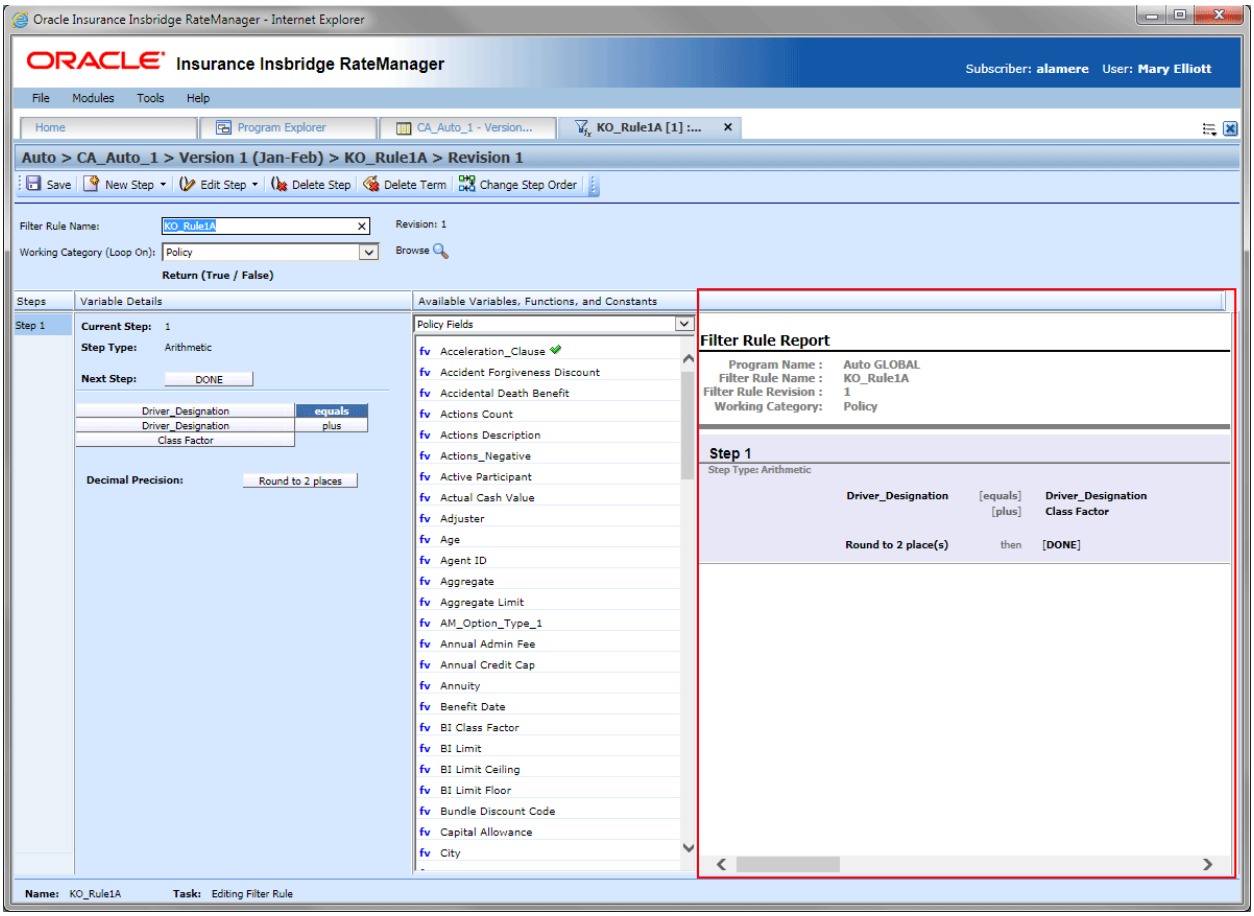


Figure 430 Filter Rules Report

CREATING A FILTER RULE

Filter rules can be created and added to standard programs and template generated programs. Local filter rules are created at the program level and are available to local elements only. Local filter rules can be copied to other programs within the project. Global filter rules are created in the globals area of Program Details or on the globals tab and can be used by global or local elements. Global filter rules can be copied to other programs within the project. Universal filter rules can only be created on the universal tab but can be used by universal, global or local elements.

Local and global filter rules are created and function in much the same way.

1. To create a universal filter rule, select the universal icon under the project where you want to create a filter rule on Program Explorer.

2. On the Program Details tab of the program where you want to create a new filter rule, select either the Global Filter Rules folder or the Local Filter Rules folder.

3. Click New in the top bar menu. If you do not select a folder, you can select New→Globals→Filter Rules or New→Locals→Filter Rules.

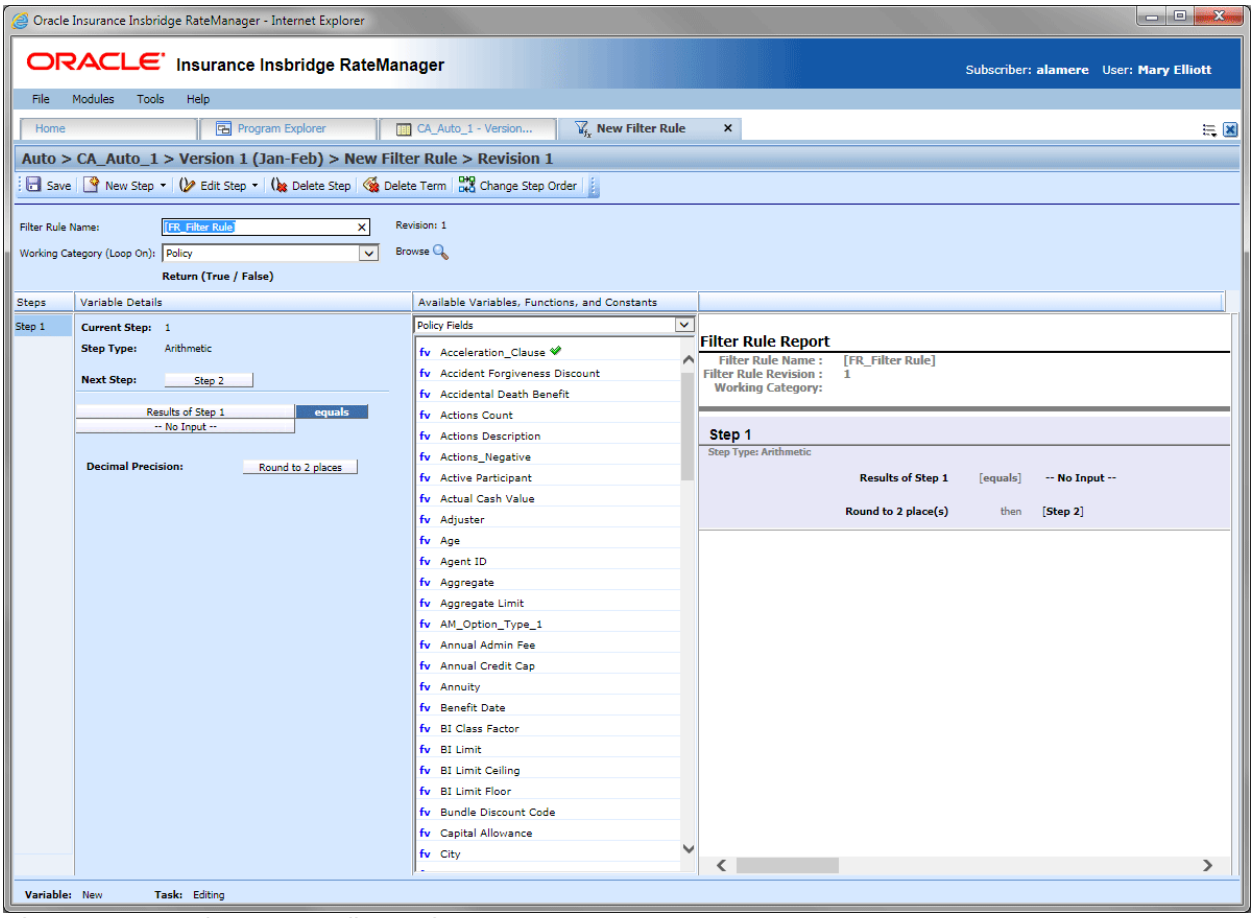


Figure 431 Creating a New Filter Rule

4. This opens the New Filter Rule tab for the new rule where you can enter or select the variable elements you need.

5. Enter a Name for the filter rule. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

6. Choose a Working Category.

   You can select the working category from the menu or you can browse for a working category by clicking BROWSE to bring up the Browse Category window.

   From here you can navigate to the Category you want. Once you have selected the category, click OK. The category auto fills in the working category section.

   If this is not the screen you want, click Cancel to close the window and return to the previous screen.

NOTE: *Always save a filter rule after defining the filter rule name, and working category. The filter rule will not be available to be used within a step until the filter rule has been saved.*

7. The first step is populated in the Variable Details area. This must be a mathematical step. If you do not want an arithmetic step as your first step, add the desired step and then delete the arithmetic step. To do so, highlight Step 1 **and click** Delete Step in the menu bar at the top of the tab.

8. To add a new step, click New Step and select the type of step you want to add. For more information on the types of steps, see Working with Steps.

9. The screen refreshes with the new step parameters. Depending upon the step type, you can select the appropriate variables, constants or fields. The values required are listed on the step.

   For example, if the step has an “Enter Variable” box, you must choose a variable from the menu in the Available Variables, Functions and Constants area. The options are populated underneath. Click the “Enter Variable” box and then double-click the corresponding variable you want. The box refreshes with your selected value.

10. Continue entering steps until complete.

11. The last step in the filter rule must specify if the rule should return a true or false value.

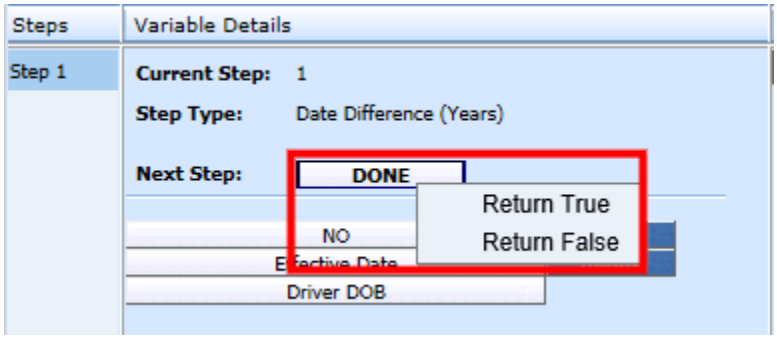


Figure 432 Selecting a True or False

12. For the Next Step selection, select whether you want to return a true or false value. The value selected determines if the rule is run, for example, if the filter rule is true, the variable or algorithm is run.

13. When you are finished making your entries, click Save.

Adding a Variable or a Custom Value

A new field variable, or custom value can be created while entering or editing a filter rule without leaving the Edit tab.

- To add a field **Variable**: See Adding a New Field from Another Area
- To add a **Custom Value**: See Custom Value

EDITING A FILTER RULE

The edit filter rule tab is where you can change information such as name, working category and steps. If the filter rule is locked, no changes are allowed.

Filter rules that are locked cannot be edited. Template generated filter rules that are using revision 1 are not open for editing. If you want to edit a template generated filter rule, you must create a new revision. The new revision is open for editing except for the name. The name cannot be changed.

NOTE: *If a change is made to a global filter rule , the change affects all programs that use the global filter rule. If a change only needs to be made to one particular program, create a local filter rule in the program and have the program reference the local filter rule instead of the global.*

Editing a Filter Rule

1. On the Program Details tab, select the filter rule you want to edit.
2. Double click or right click and select **Edit**. The Edit Filter Rule tab opens with the details of the rule.
3. Make any changes.

   To change the step order, you have two options:

   - **Option 1:** Select a step in the Steps column, drag it to a new location in the list and drop it.
   - **Option 2:** Select the step you want to move and click CHANGE STEP ORDER. A separate screen is displayed.
     - The current step position is displayed at the top of the screen. Select where you want the step to be placed.
     - Click OK. You will be returned to the edit filter rule tab. The step should be placed where you chose.

   To delete a step, select the step in the Steps column and then click DELETE STEP. You cannot delete the last step from a filter rule.
4. Click **Save** to save your work.

WARNING: Category Updates to Revisions

Category updates should not be done to a revision. All revisions of a filter rule should be at the same category level. Changing the category for only some of the revisions of a filter rule is not supported.

COPYING A FILTER RULE

Filter rules may be copied in any program. Local filter rule can be copied to the same program or to another program within the same project at the local level. Global filter rules can be copied to the same program at the global level or, to the same program at the local level, or to another program within the same project at the local level. Universal filter rules can only be copied at the universal level.

Variables may be copied in any program.

- **Local variables** can be copied to the same program or to another program at the local level within the same product.
- **Global variables** can be copied:
  - At the global level and are available to every program in the product
  - To the local program level
  - To another program within the same product at the local level
- **Universal variables** can be copied:
  - At the universal level and are available to every program in the project
  - To the local level of any program within the project

The copy options depend on whether the variable to be copied is at the universal level, global level or the local level.

Universal Copies

When copying a universal variable to a local level:

- Universal copies to a local level are project wide. Information has been added to help you select the Destination Program where you want to copy the element to. The Destination Program name is listed first, the folder name is in parenthesis and the product name is last, also in parenthesis.
- Duplicate naming is not allowed. The New Name must be different from the Old Name.
- Product level naming standard are not enforced. If the product has a naming standard, copies made from the universal level are not required to have the naming standard applied when copying. If the variable is edited any time after the initial copy, the naming standard is enforced.
- Universal permissions are required to make copies at the universal level.

Global Copies

A global copy creates a copy of the filter rule and places it at the global level. There are no options.

When copying a global variable to a local level:

- Global copies are done at the product level. The Destination Program name is listed first and the folder name is in parenthesis.
- The New Name must be different from the Old Name.

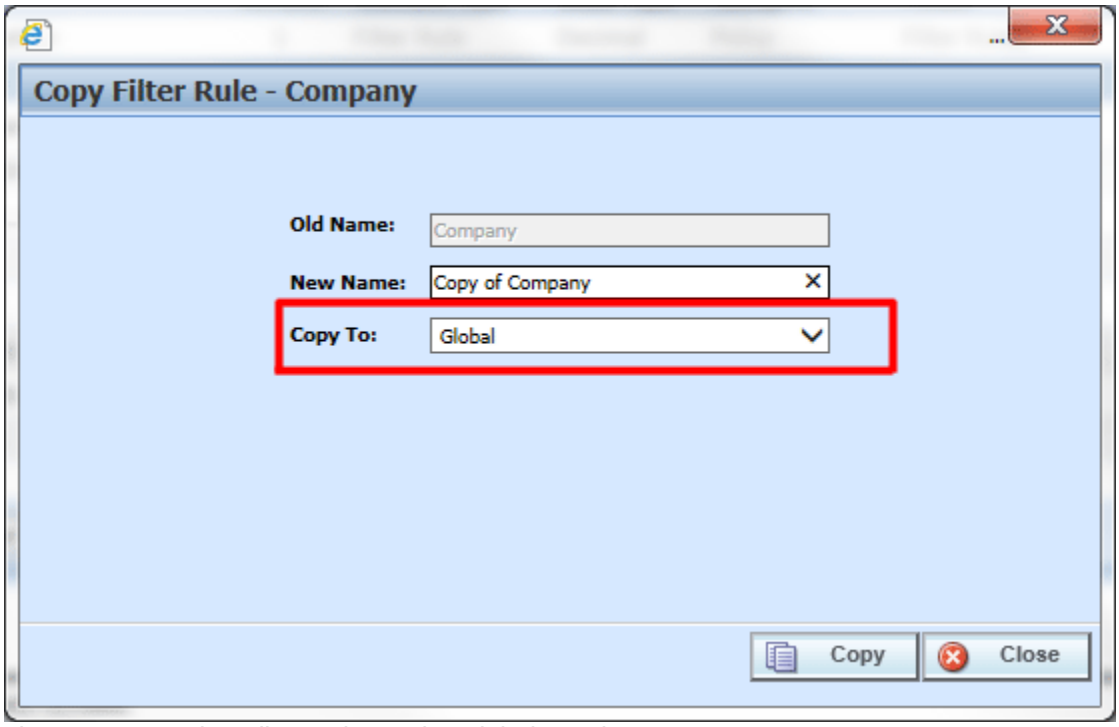


Figure 433 Copying Filter Rules at the Global Level

Local Copies

When a global filter rule is copied to a local level or when a local filter rule is copied, there are two copy options:

- **Existing Program:** Copy the filter rule into the existing program under another name.
- **Other Program:** Copy the filter rule into another program with the same name. This option is for filter rules within your default project.

Local copies can be done to the local level of any program in the product. When copying a variable to a local level in another program:

- Local copies are done at the product level. The Destination Program name is listed first and the folder is in parenthesis.
- Local to local copies check for dependencies when the copy is made:
  - When performing a local to local copy, dependencies such as the usage of other local elements are checked. If there are local elements used in the element being copied that are not in the Destination Program, then those elements are copied also.

  - For example, the Option AB Fee algorithm, a local variable, uses variables Option A and Option B, also local variables. When Option AB Fee is copied from program Maine to program Kansas, the algorithm is copied and so is the variables that are being used.

- Local to local copies do not require a name change:

  - When copying to the local level of a different program, the variable name cannot be changed. The New Name is grayed out. This creates an identical variable in the other program and reduces the need for editing. If a change is made to the original variable and then the variable is copied for a second time, the copy is created as a new revision in the Destination Program.

  - When copying at the local level in the same program, the name must be changed.

- Copies made in the current program must have a new name.

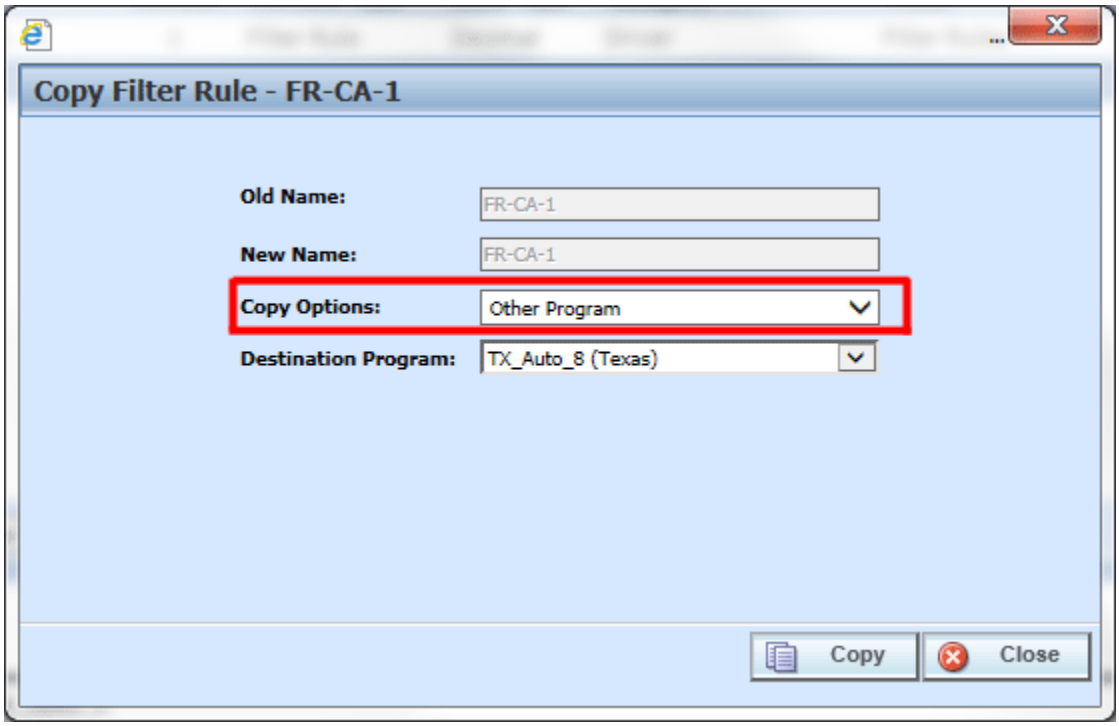


Figure 434 Copying Filter Rule at the Local Level

COPYING A FILTER RULE

1. Select the filter rule you want to copy and right click. Select **Copy** from the menu.

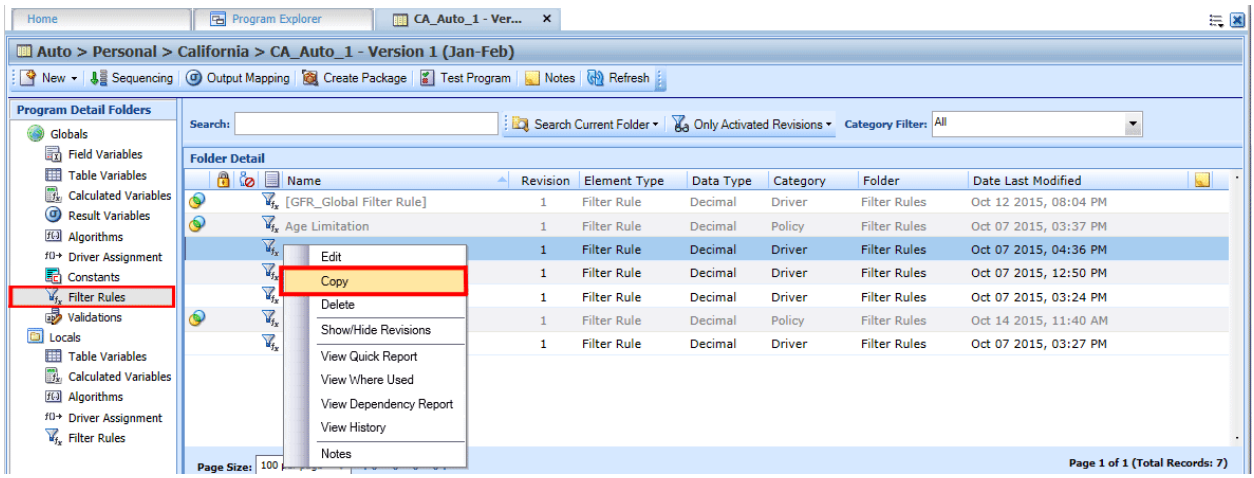


Figure 435 Copy Filter Rule Menu

2. A separate window is displayed.

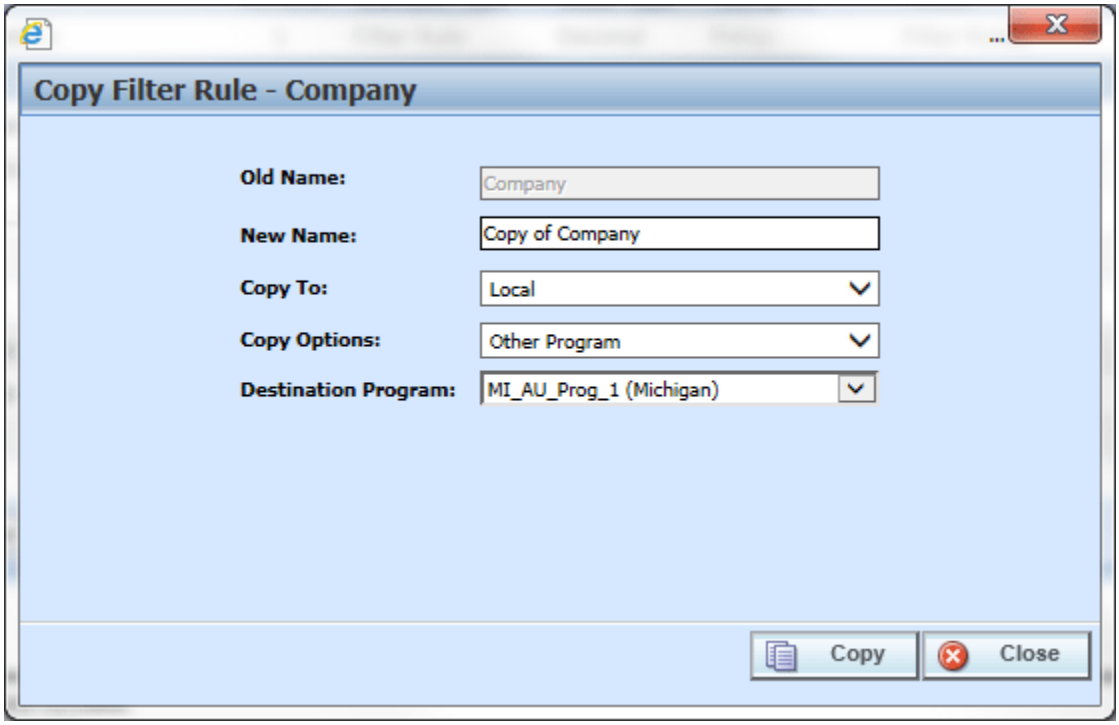


Figure 436 Copy Filter Rule Screen

3. Select the **Copy Options**. It is recommended that you change the copy options first. Changing options may reset the name field.

4. If this is a global filter rule, select if this is a local or global copy. If this is a local filter rule, you can select the Other Program where you want to copy to. This option is not on universal filter rules.

5. If allowed, enter in a **New Name**. Local copies to another program cannot change the name. The name can be entered first. It is recommended that you change the copy options first. Changing options may reset the name field.

6. Click **Copy** to copy the filter rule. Dependencies are searched. If there are any dependencies that need to be resolved, a window is displayed. See Resolving Dependencies.

7. If the copy was successful, you will be returned to the previous screen. Your copied filter rule is listed. The last updated date is the date the copy was made.

DELETING A FILTER RULE

When a filter rule is no longer needed, it can either be left in RateManager or deleted. When you delete a filter rule, it gets deleted from all versions of the program. Also, all revisions of that filter rule get deleted. If you are uncertain as to whether the filter rule will be used again, simply leave it in RateManager. This will save time if the filter rule is needed and won't slow down rating.

To Delete a Filter Rule

1. Select the filter rule you want to delete. Right click and select **Delete**.

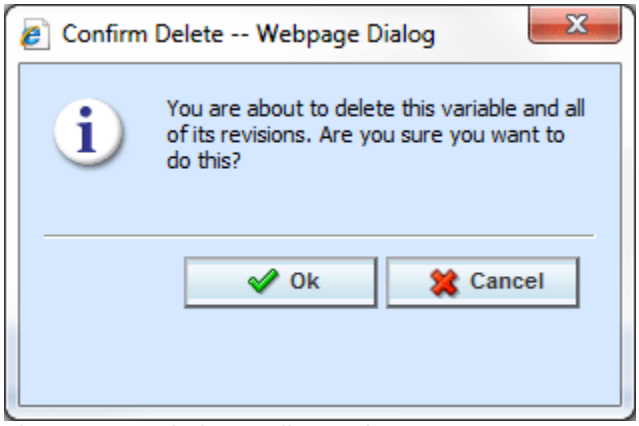


Figure 437 Deleting a Filter Rule

2. You will be asked to confirm the deletion of the filter rule. Click **Ok** to delete the filter rule. Click **Cancel** to return to the tab without affecting a change.

If you try to delete a filter rule that is still being used by an algorithm, a result group or another variable, you will receive an error. Filter rules that are locked or that are template generated cannot be deleted.

NOTE: *Filter Rules that are in use cannot be deleted.*

Figure 438 Error Message for Deleting a Filter Rule

Adding a Filter Rule to a Variable

Filter rules can be added to table variables, calculated variables, or algorithms. The Select Filter Rule option is located in the Variable Information section of each of these elements. The procedure for adding a filter rule is the same.

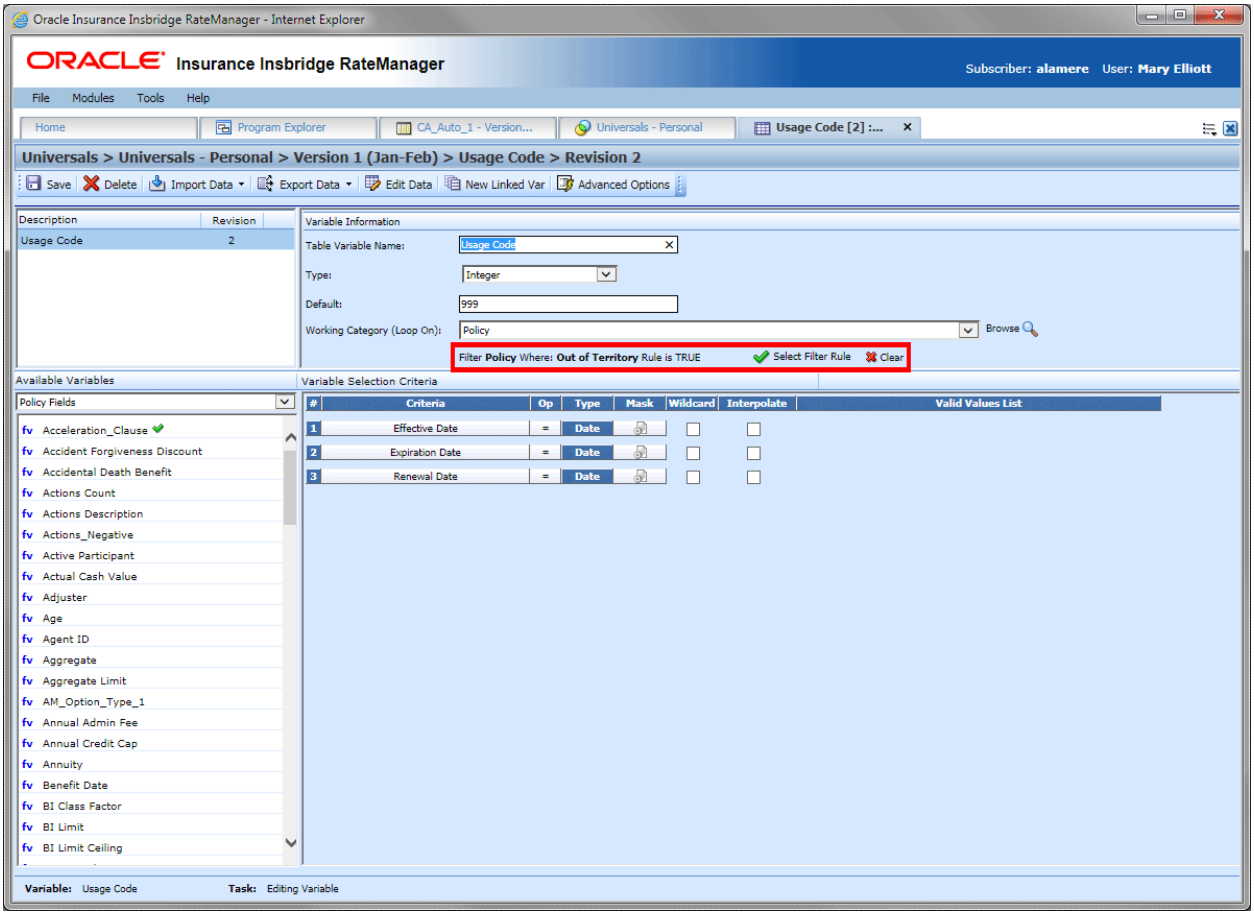


Figure 439 Adding a Filter Rule to a Variable

There are three fields for Select Filter Rules options:

Filter **Policy** Where: Filter **Policy** Where: **Out of Territory** Rule is TRUE	**Filter Policy Where:** – If blank, no filter rule has been applied. – If populated, the filter rule name is displayed as well as when the rule is to be run. If the rule is TRUE, then the variable or algorithm will run if the conditions of the filter rule are met. If the conditions of the filter rule are not met, the variable or algorithm is not run.
Select Filter Rule	**Select Filter Rule:** Click this option to open the selection screen.

Clear	**Clear:** Clears the filter rule. The action is immediate. No warning message is presented.

Filter rules can be applied to universal, global or local table variables, calculated variables, or algorithms.

- Universal elements can only have universal filter rules applied.
- Global elements can have either universal or global filter rules applied.
- Local elements can have universal, global or local filter rules applied.

Only one rule can be applied. If a filter rule is already in place, selecting another replaces the existing one with the new one.

When the Select Filter Rule is clicked, the Select Filter Rule – For Category is displayed.

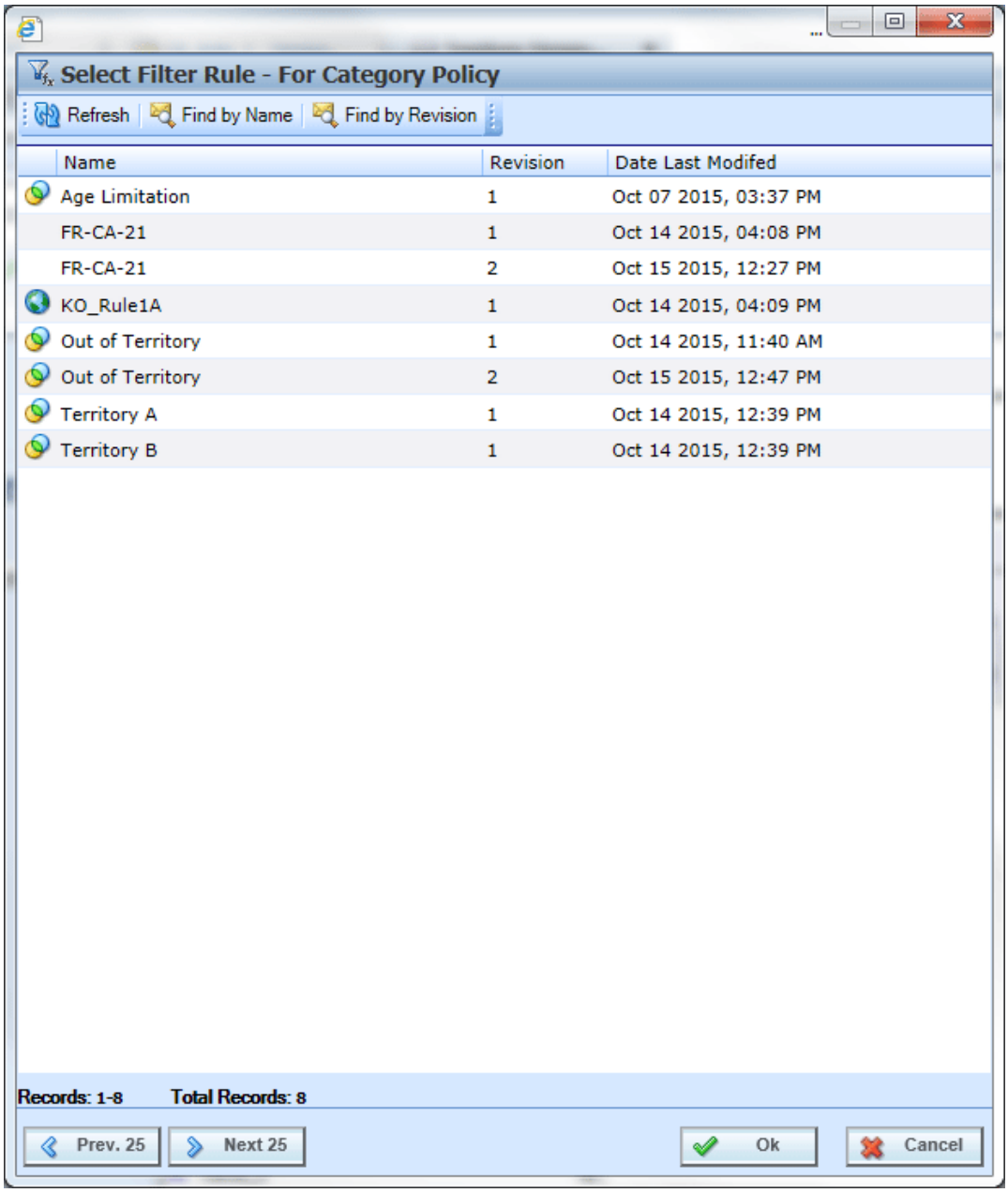


Figure 440 Select Filter Rule by Category

Select Filter Rules Details

The select filter rules screen contains a listing of the filter rules available for the Working Category. The filter rules are further defined by the location of the element. For example, if this is a universal element, only universal filter rules are displayed. Local elements have filter rules for universal, global or local.

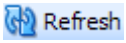
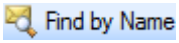
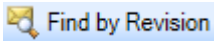
Filter rules are listed alphabetically with the total number of filter rules displayed on the bottom of the screen. Paging options are available when the listing exceeds 25.

The listing can be sorted by name, revision or date. Click the title that you want to sort on to view filter rules in that order.

- **Name:** A unique, user defined name for the filter rule. Names cannot be duplicated by any other filter rule.
- **Revision:** A system assigned revision number.
- **Date Last Modified:** The date the last modification was made to the filter rule.

There are no filter rule right click menus.

Select Filter Rules Options

	**Refresh:** Refreshes the list of filter rules.
	**Find by Name:** Searches the name field in the currently displayed filter rules for the value entered in the search field and returns with a list of matching results. Searches are case sensitive.
	**Find by Revision:** Searches the revision field in the currently displayed filter rules for the value entered in the search field and returns with a list of matching results. Searches are case sensitive.
	Universal: Indicates that the filter rule was created at the universal level.
	Global: Indicates that the filter rule was created at the global level.

Adding a Filter Rule

1. Open the variable where you want to add a filter rule.
2. In the variable information section, click the Select Filter Rule option. This opens the select Filter rule by category.
3. Select the filter rule you need and click OK.
4. On the variable information section, the filter rule is listed. I filter icon is at the front of the element name.

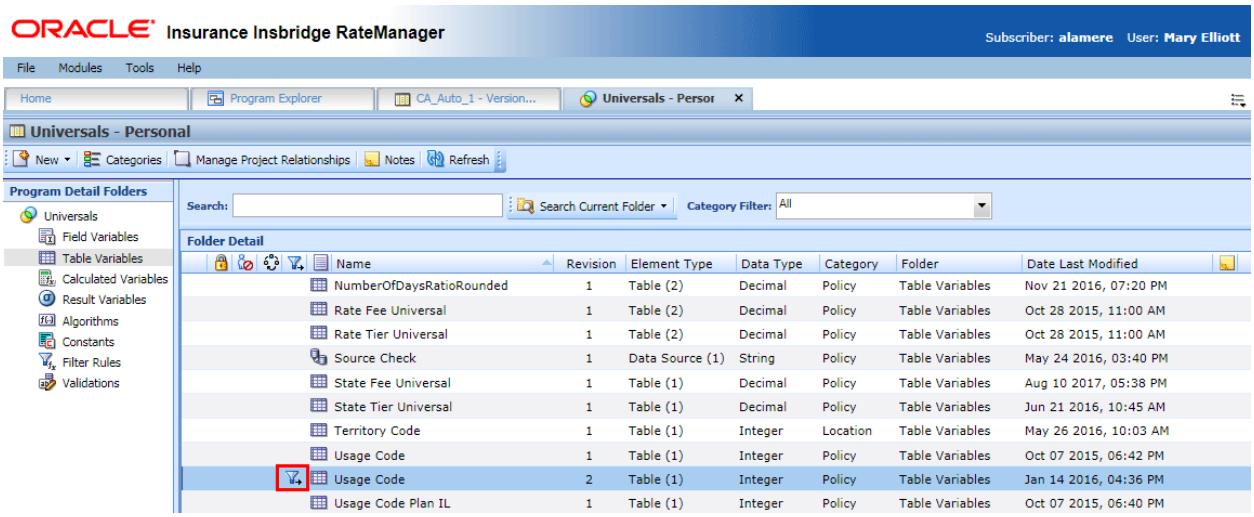


Figure 441 Universal Table Variable with a Filter Rule

5. The rule can be removed by clicking the clear option on the Edit Variable tab.

NOTE: *If the variable or algorithms changes categories, the filter rule is removed. Filter Rules must be selected from the working category.*

CALLOUTS

Callouts are programs or SoftLibraries that are used by the initiating program to run another program or call to an outside party. A callout is a step within a local algorithm or driver assignment scenario. When the algorithm or driver assignment is executed, the callout runs the selected program or SoftLibrary and returns with results that can be carried further into the rating.

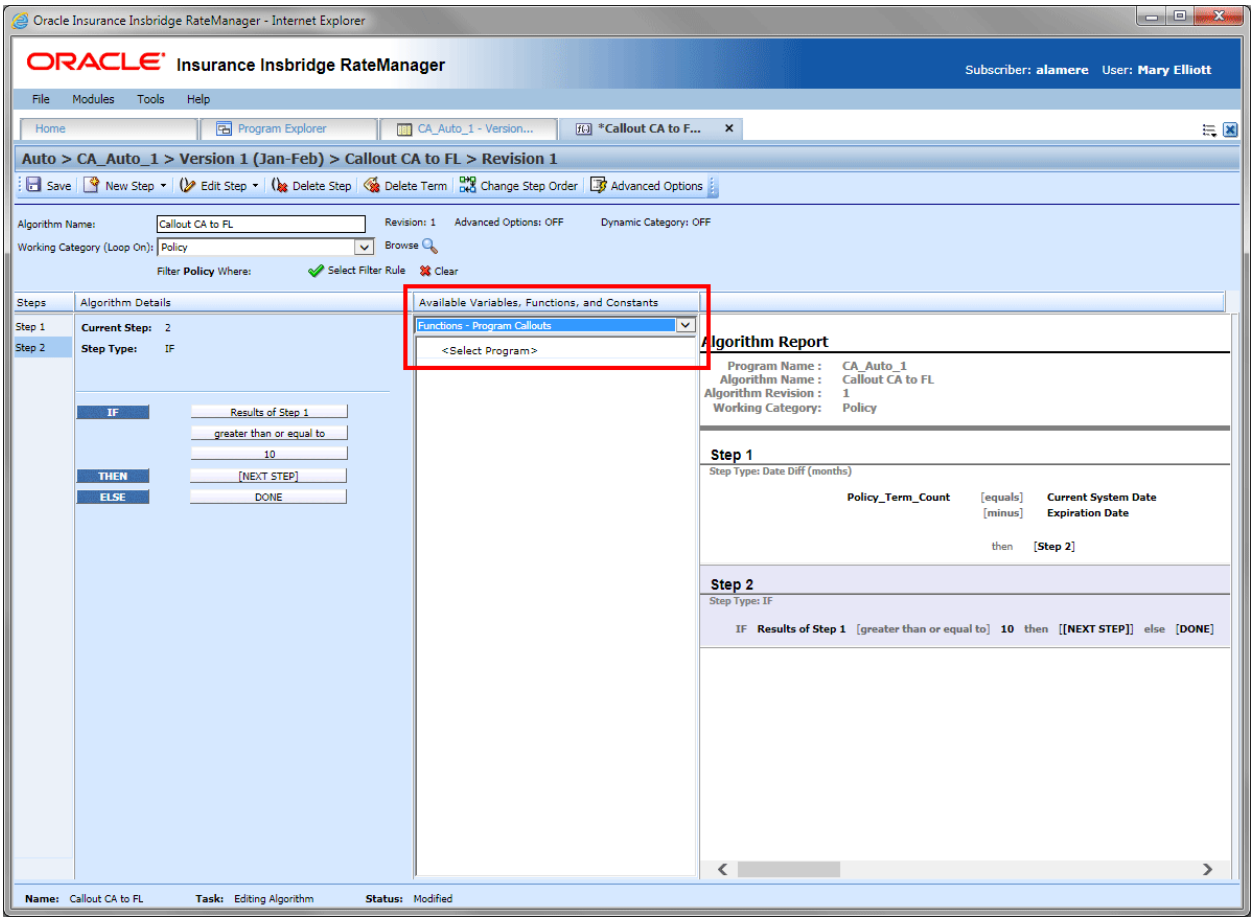


Figure 442 Callout Algorithm

Callouts can call either another program or a library:

- **P2P:** Program to Program Callout. When one program makes a call to another program and requests that program to run. P2P callouts can be made to a program in the same project, or across a domain in to another project.
  - **Same project:** When a program calls another program within the same project.
  - **Cross project:** When a program calls another program in a different project in the same domain or another program in a different project in a different domain. Cross project relationships must be set up prior to making the callout.

- **P2L:** Program to Library callout. When a program makes a call to a SoftLibrary that has been loaded into IBFA. A SoftLibrary is a specially developed program that performs a specific task. SoftLibraries may run their own code or call upon other systems to obtain information outside of RateManager, for example, obtaining a credit score.

NOTE: *SoftLibraries are loaded into the IBFA by a system administrator. SoftLibraries are displayed in RateManager only after they have been loaded into IBFA. If a SoftLibrary you need is not available, please contact your system administrator to have it added.*

Within a program, you can have one or more P2P callouts, one or more P2L callouts or any combination of P2P and P2L callouts. There is no limit to the number of callouts you can use.

SAME PROJECT P2P CALLOUTS

Same project callouts are used to call one or more programs within the same project and domain. Same project callouts use the same universal variables. Available fields and categories are identical within the programs. No mapping is required. If there is an exception, you can create a mapping for it.

- This feature is available to any program within the project.
- No mapping is required. Mapping is automatically matched up.
- Same project callouts can be tested in Test Case Editor. All fields are displayed.

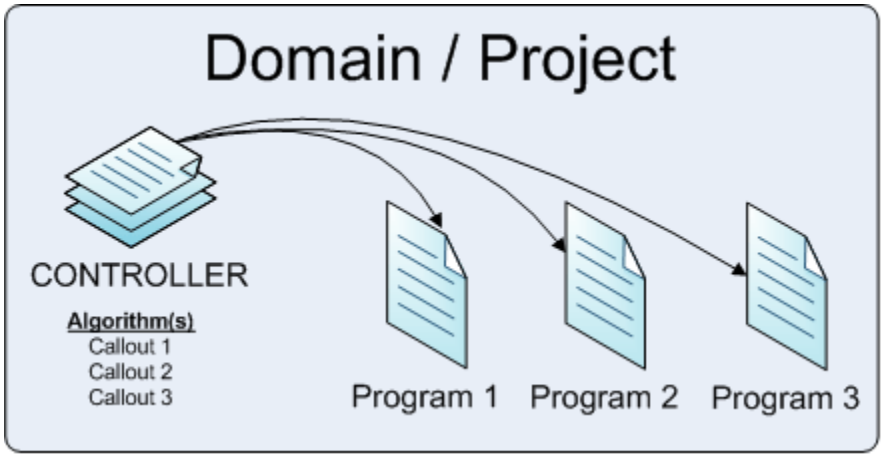


Figure 443 Same Project P2P

For example, you can set up special programs that run specific functions (Program 1, Program 2, and Program 3) and then create a master program that calls the smaller programs (Controller) when needed. All programs are created in the same project using the same categories and fields.

The special programs are created like any other program. The controller program contains the algorithm(s) that performs the callouts. Callouts are steps within an algorithm. You can create one algorithm with three callout steps or three separate algorithms with one callout step per algorithm.

CROSS PROJECT CALLOUTS

Cross project callouts allow for a program to callout to another program in a different project in the same domain or in a different domain. Each project can set up a single relationship with one other project per domain in the Manage Project Relationships window.

For example, in the Alamere domain, the Personal project can establish a relationship with the Training domain, Project Training project. The Alamere domain cannot set up any other relationships with any other project. The Alamere domain can be edited to form a new relationship with a different project. The relationship with the Project Training project would not remain.

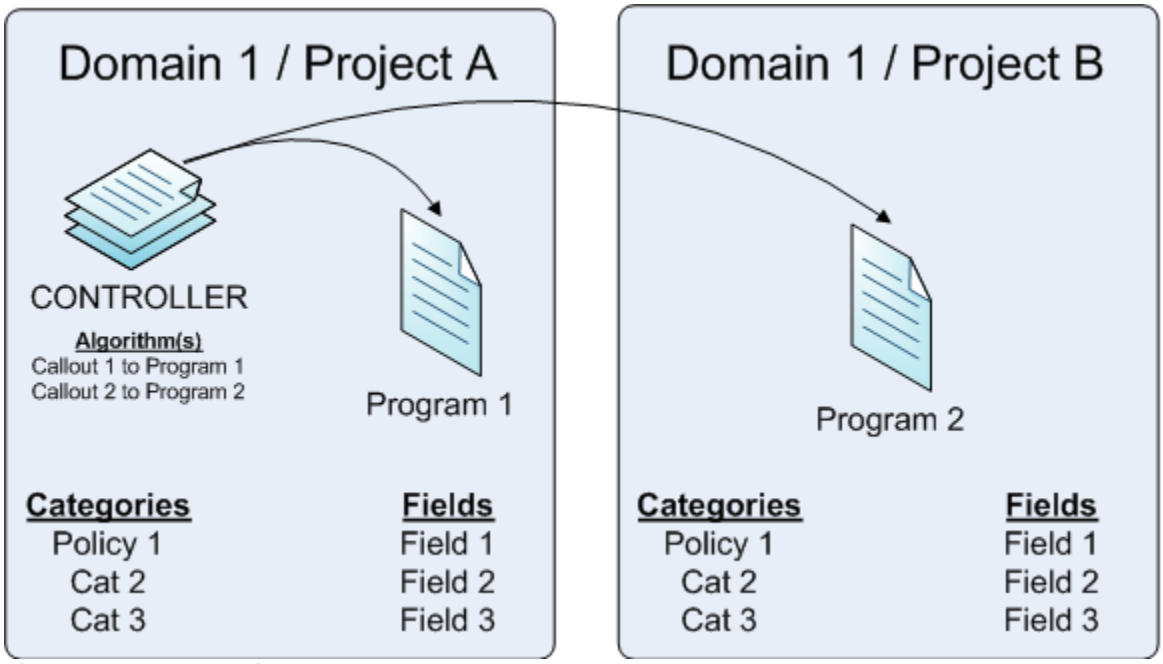


Figure 444 Cross Project P2P

For example, you can set up special programs that run specific functions (Program 1 and Program 2) in one or more projects. Then create a master program in that calls the smaller programs (Controller) when needed. Even though the programs are created in different projects, they use the same categories and fields because they were all created under the same domain.

The special programs are created like any other program. The controller program contains the algorithm(s) that performs the callouts. Callouts are steps within an algorithm. You can create one algorithm with two callout steps or two separate algorithms with one callout step per algorithm.

- Cross project relationships are needed if you want to make a callout to a program in another project. Relationships must be granted prior to creating any callouts. Until a relationship is granted, the option to callout to another project will not be available.
- A cross project callout requires that a relationship be established with the called project. Please see Manage Project Relationships for more information.
- Cross project callouts must have matching schemas. Categories and fields must have identical XML IDs for the elements that are to be passed back and forth. If categories are not created identically, then unexpected results may occur.

Result Variables

When a cross project program is utilized, the called programs result variables are brought in to the calling programs result variables tab when the cross project relationship is created. The result variables are displayed with the P2P abbreviation found on the Select Product screen of the project followed by an underscore (_). For example, if the called program is in the Alamere Project with the P2P abbreviation of ALA, any result variables listed on the calling programs result tab look like this:

ALA_*ResultVariableName*

Where: ResultVariableName is the name of the result variable brought in.

If the project does not have a P2P abbreviation, then the result variables will begin with an underscore (_).

Result variables are brought in for every category that matches. If there are categories in the called program that are not in the calling program, those results will not be brought in or available.

Result variables from the called program are available for editing or usage in the calling program.

USING P2P CALLOUTS

P2P callouts are used in steps in local algorithms or driver assignments. To use a callout, select the function – Program Callouts from the Available Variables, Functions, and Constants. The available programs are listed on a separate window. Select the callout that you want to run. This populates the step in the algorithm or driver assignment. You also can select a mapping and a program version. These are optional and not required.

If you choose to have multiple programs in an algorithm, please be aware that a program cannot make a callout back to the initiating program. This is circular logic and will result in an error.

For callouts, step types do not matter. The entire step consists of the callout. No other entry can be made. Callouts are program specific and are available in local algorithms and driver assignments only.

Prior to using a P2P callout you should have ready:

- **Mapping Exceptions:** If required. A mapping exception is when the field and result variables do not map automatically and require manual mapping or when a specific field or result variable needs to be mapped to another specific field or result variable.
- **Cross project relationships:** If required. A cross project relationship allows for a program to callout to another program in a different project in a different domain. If cross project relationships have been created, those will be listed. If only programs from the current project are listed, cross project relationships have not been activated.
- **Program Version Variable:** If required. If you want to call a specific program version, you can select a version using a version criteria variable. This requires a version criteria variable to already be selected at the program version level.

All programs in the project are listed when selecting a program to be called. Callouts require that the called program be packaged. If you select a program that has not been packaged, you may receive an error.

P2P MAPPING EXCEPTIONS

Mapping exceptions map the field and result variables of the calling program to the called program. Callout mappings are needed only for:

- In P2P callouts within the same project, mapping is automatically matched up. Exceptions require special mapping and multiple exception mappings can be created if needed.
- Exception mappings are done at the program level and are available to any program version.

A P2P callout that requires a mapping is a mapping exception. A mapping exception is when the field and result variables do not map automatically and require manual mapping or when a specific field or result variable needs to be mapped to another specific field or result variable.

For example, if there is a field variable called garage ZIP in the calling program but in the called program the same field value is required in a variable called parking postal code. If the fields are not automatically mapped, then a mapping exception is needed.

Callout mappings are done at the program level and are available to every program version.

Results mappings are at the same level. You cannot map a universal result variable to a global result variable.

At this time, the driver/vehicle category found in Drivers Assignments cannot be mapped.

Creating a P2P Exception Mapping

1. Navigate to the domain type→domain→project→folder→program where you want to map a callout.

2. Right click it and select **Map Callouts** from the menu.

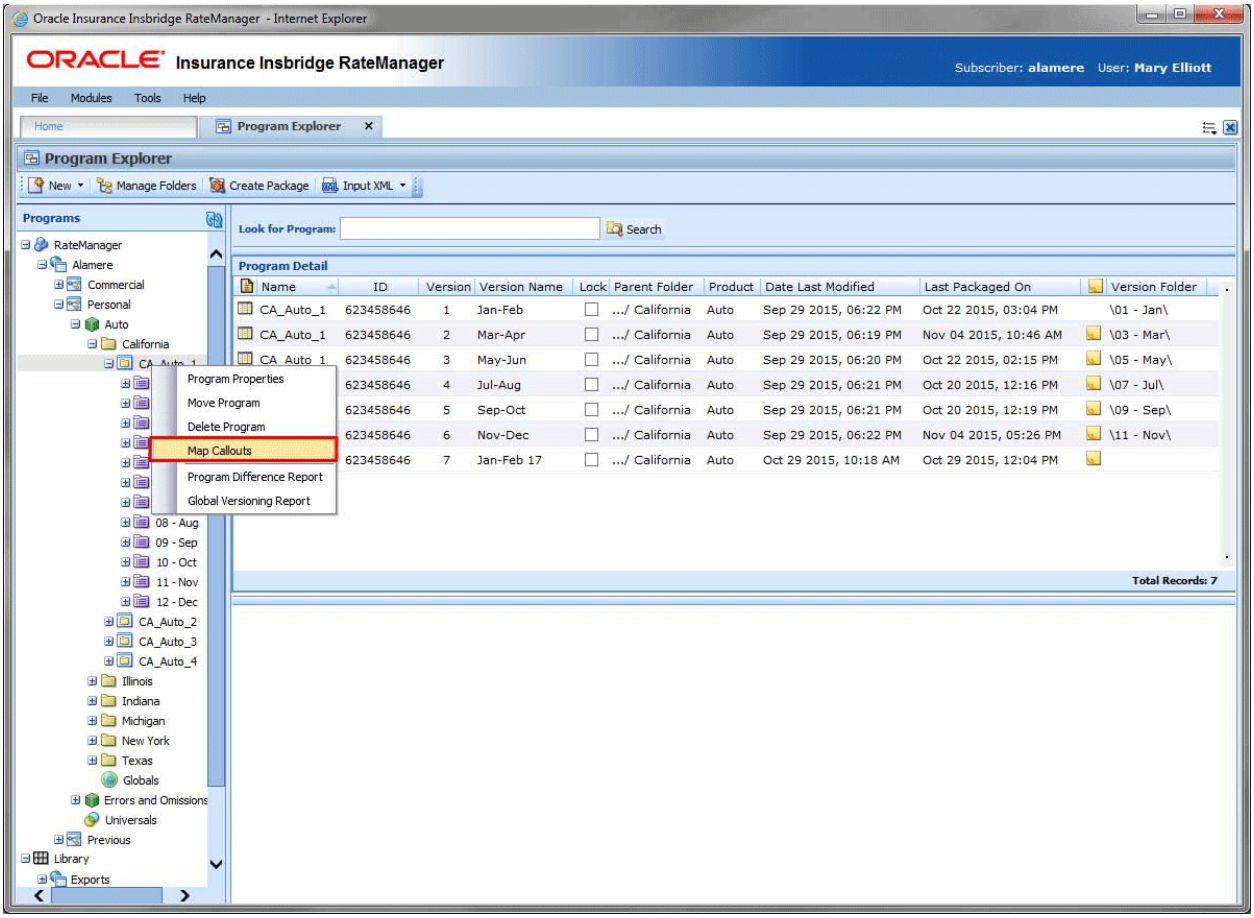


Figure 445 Navigating to Callouts

3. The Map Callouts screen is displayed.

MAP P2P CALLOUTS

At this time, the driver/vehicle category found in Drivers Assignments cannot be mapped.

For mapping exceptions, you must set the Field and Output mappings for each program.

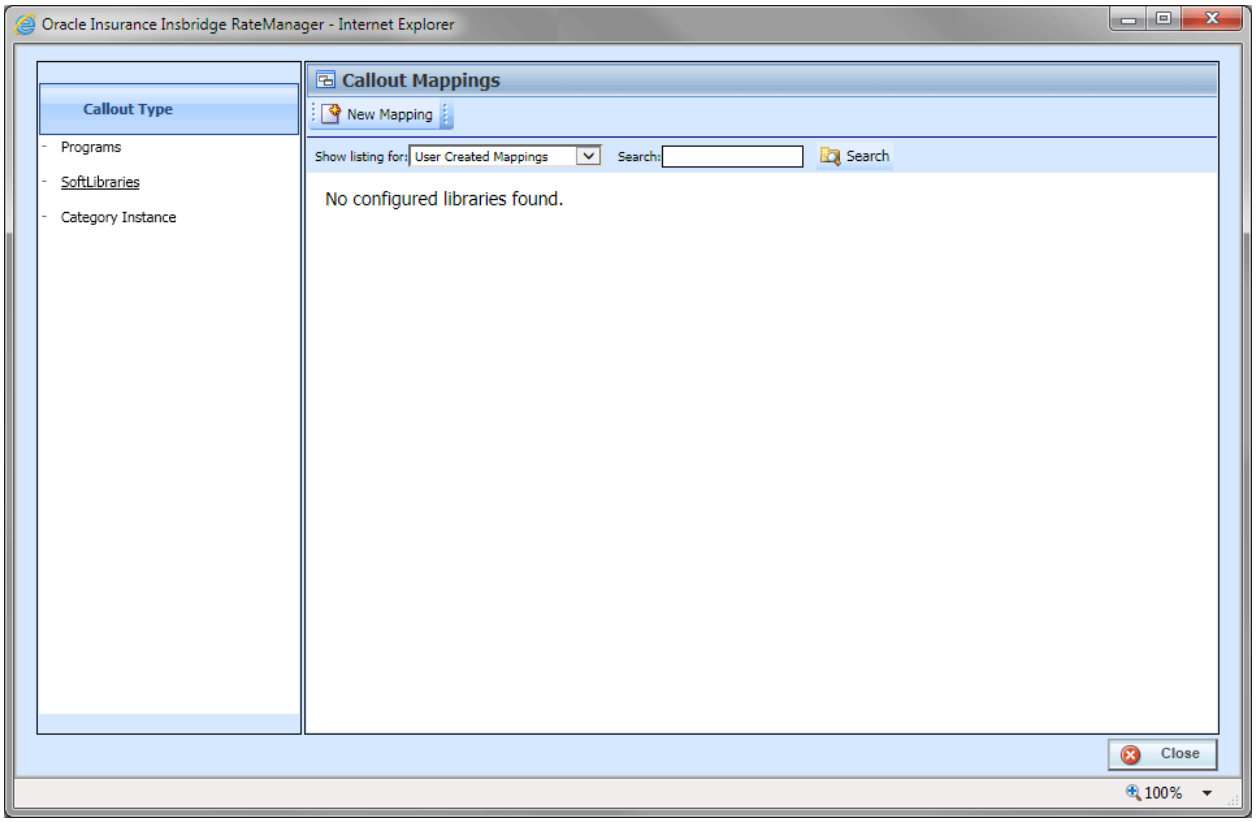


Figure 446 Map Callout Exceptions

The Map Callouts screen lists the exception mappings that have been created. There are two areas of the Map Callouts screen.

- **Callout Type:** Either Programs or SoftLibraries. The default view is Programs. To view exception mappings for SoftLibraries, click the SoftLibraries link to view mapping exceptions for SoftLibraries.
- **Mapping Listings:** Lists the currently available mappings. The default view is for recently created mappings. To view legacy mappings, change the Show Listing for option to Legacy System Mappings.

Legacy System Mappings are mappings created in previous releases. These mappings are available from the server where the Insbridge system was installed. If you are not seeing your legacy mappings, please contact your system administrator.

Legacy mappings cannot be created, only edited or deleted.

Creating a New Mapping

1. Click the New Mapping button. The program selection screen is displayed.

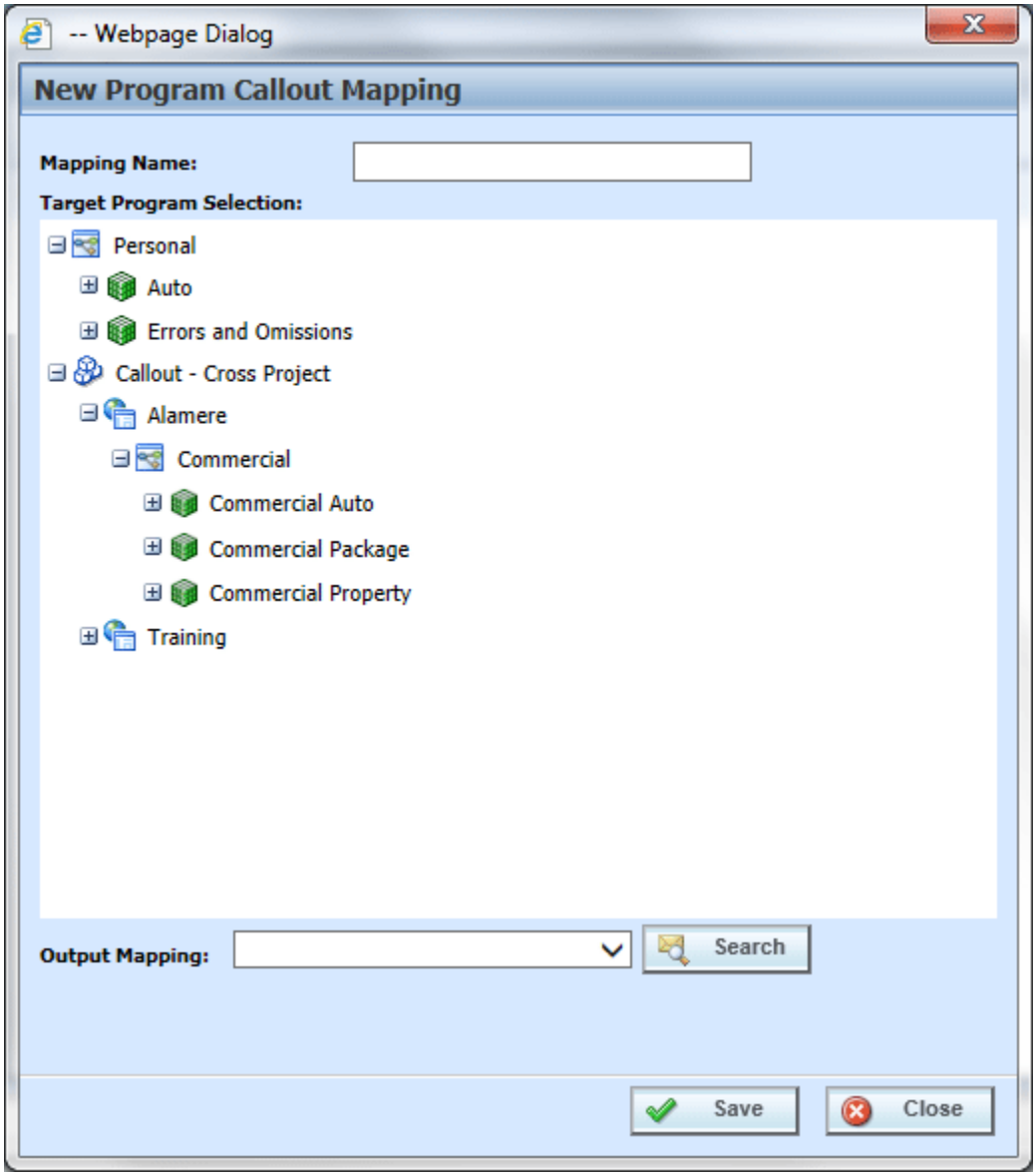


Figure 447 Creating an Exception Mapping

2. Enter a name for the mapping.

3. Select the program you want to map to.

4. Click search and select an output mapping group to use. The output mappings are located on the program being called.

5. Click Save. The new mapping is listed on the Map Callout screen.

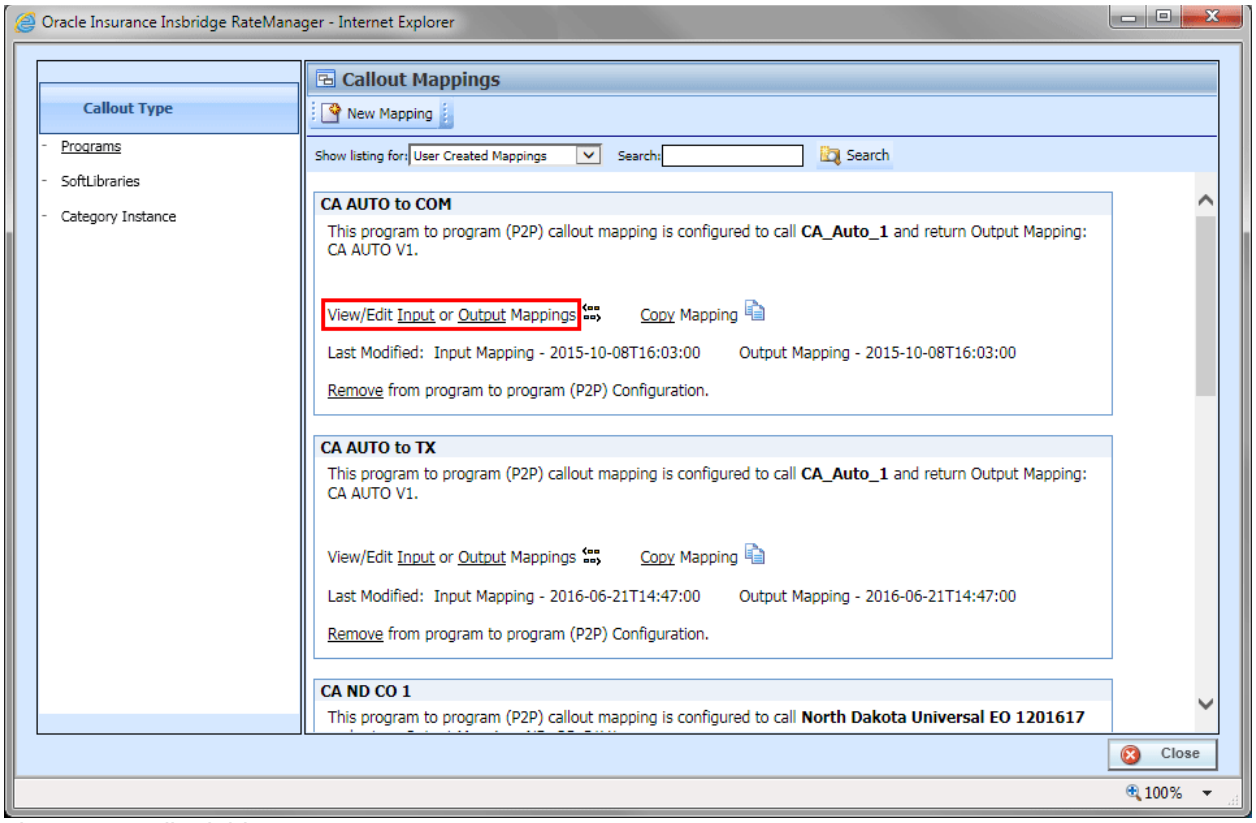


Figure 448 Edit Fields

6. To view or edit exceptions, click the Input link or the Output Mapping link. This opens the exception screen.

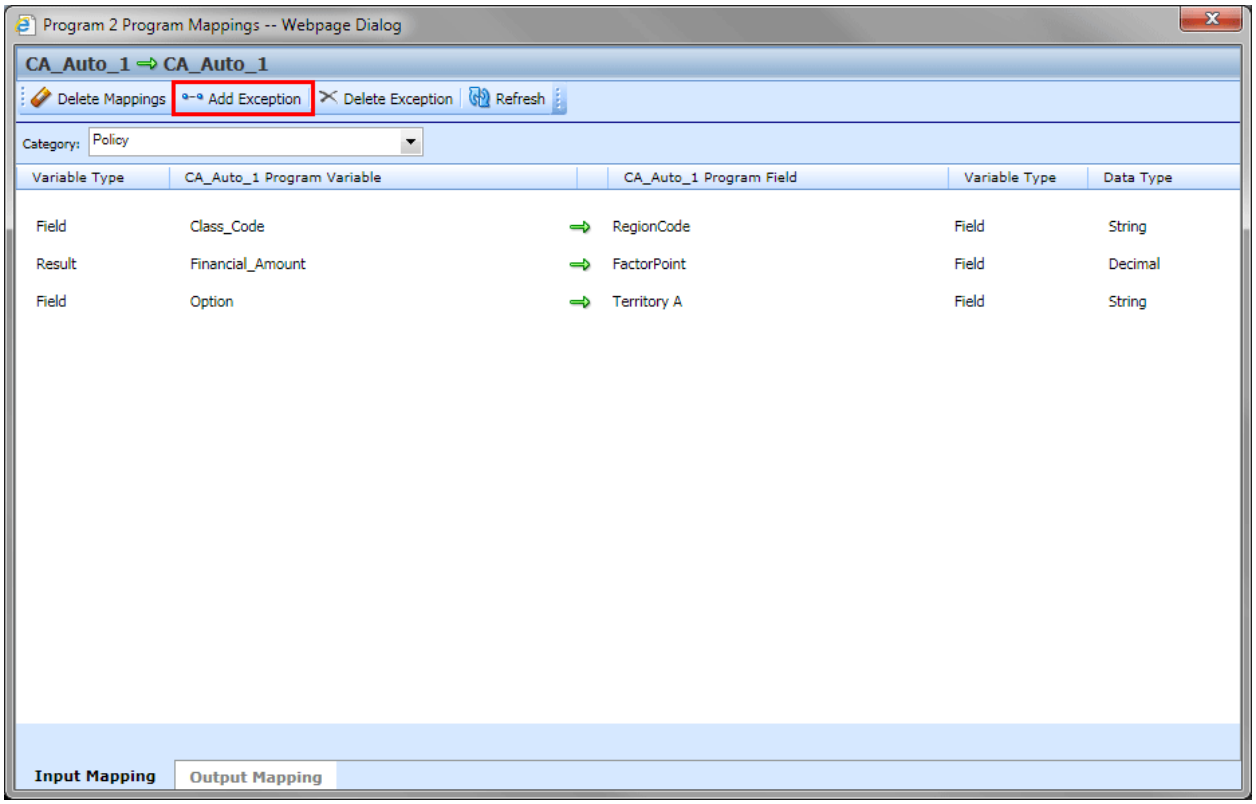


Figure 449 Adding Exceptions

7. Select the Category where you want to create an exception mapping. Policy is the default.
8. Click the Add Exception button. The exception screen is displayed.

P2P Mapping Exception Screen Functionality

The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform on this tab.

Delete Mappings	**Delete Mappings:** Deletes all the mappings. Make sure this is the action you want to take. Deleted mappings cannot be restored. They must be recreated.
Add Exception	**Add Exception:** Opens the exception mappings screen.
Delete Exception	**Map Matched Fields:** Deletes the exception. Make sure this is the action you want to take. Deleted exceptions cannot be restored. They must be recreated.
Refresh	**Refresh:** Refreshes the screen. You may need to refresh to display all exceptions.

	Direction To: Shows the direction of the exception mapping. This is for fields.
	Direction From: Shows the direction of the exception mapping. This is for outputs.

Selecting Categories on Mapping Fields

You can select the categories you want to map using the category field at the top of the screen.

Adding an Exception

When selecting the program variable, the first thirty possible choices are listed in the drop down. If the variable you need is not listed, begin typing the variable name. A list of possible selections is displayed below.

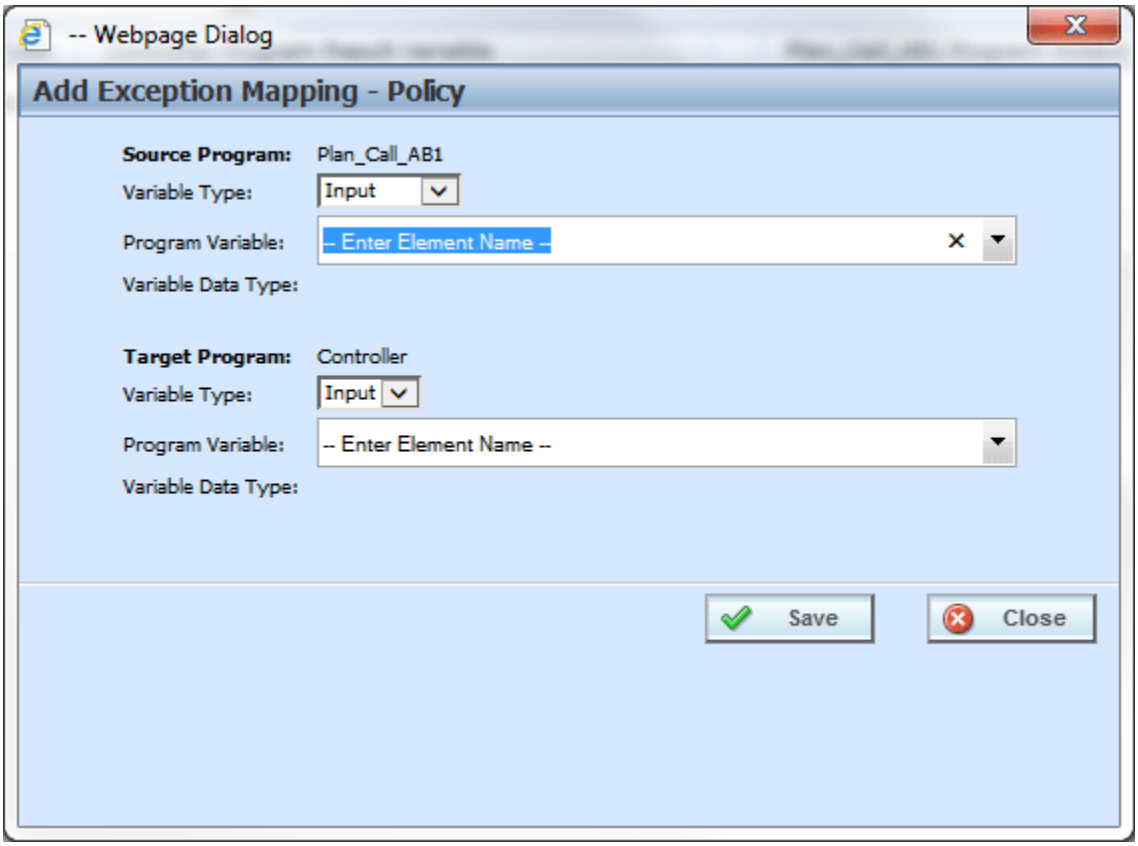


Figure 450 Exception Mapping

1. Select the variable type and the program variable from the source program. This is the program being called.

2. Select the variable type and program variable from the target program. This is the program doing the calling. The program variable types must match. You cannot map a date variable to a decimal variable.

3. Enter as many exceptions as you need. Exceptions are not listed on the exception screen until they are saved. Saving closes the screen.

4. When you are finished, lick Save. If there is a problem with the mapping, an error message is displayed. A successful mapping is listed on the exception screen.

NOTE: *While it is not required, it is recommended that the policy category be mapped first. Mapping policy variables to program equivalents before you map other categories assures that your policy fields have the closest possible match.*

Creating a P2P Callout Step

1. Select the local algorithm or driver assignment where you want to use a callout.

2. Select Program Callouts from the Available Variables, Functions, and Constants.

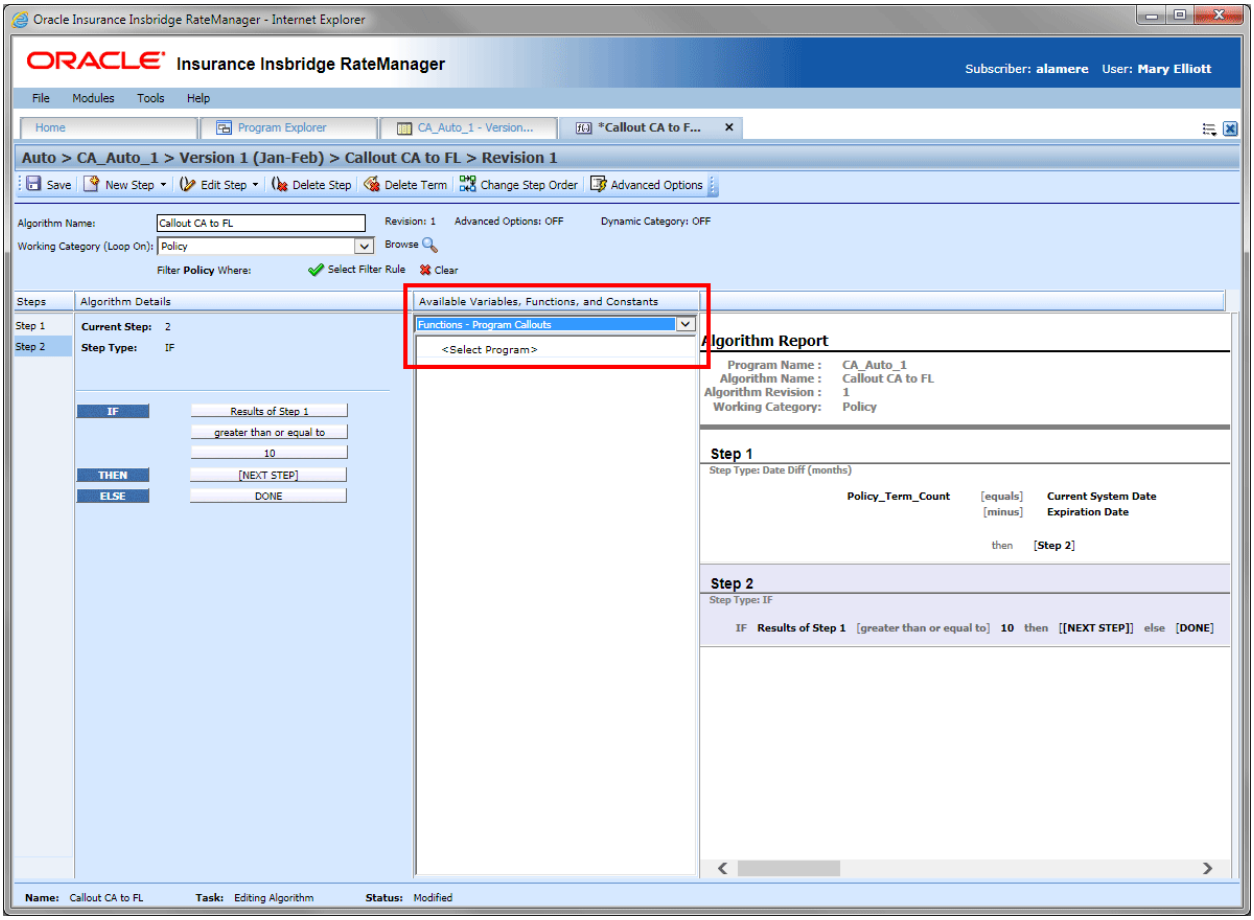


Figure 451 Selecting Functions – Program Callouts

3. Double click the <Select Program> options. A separate window is displayed.

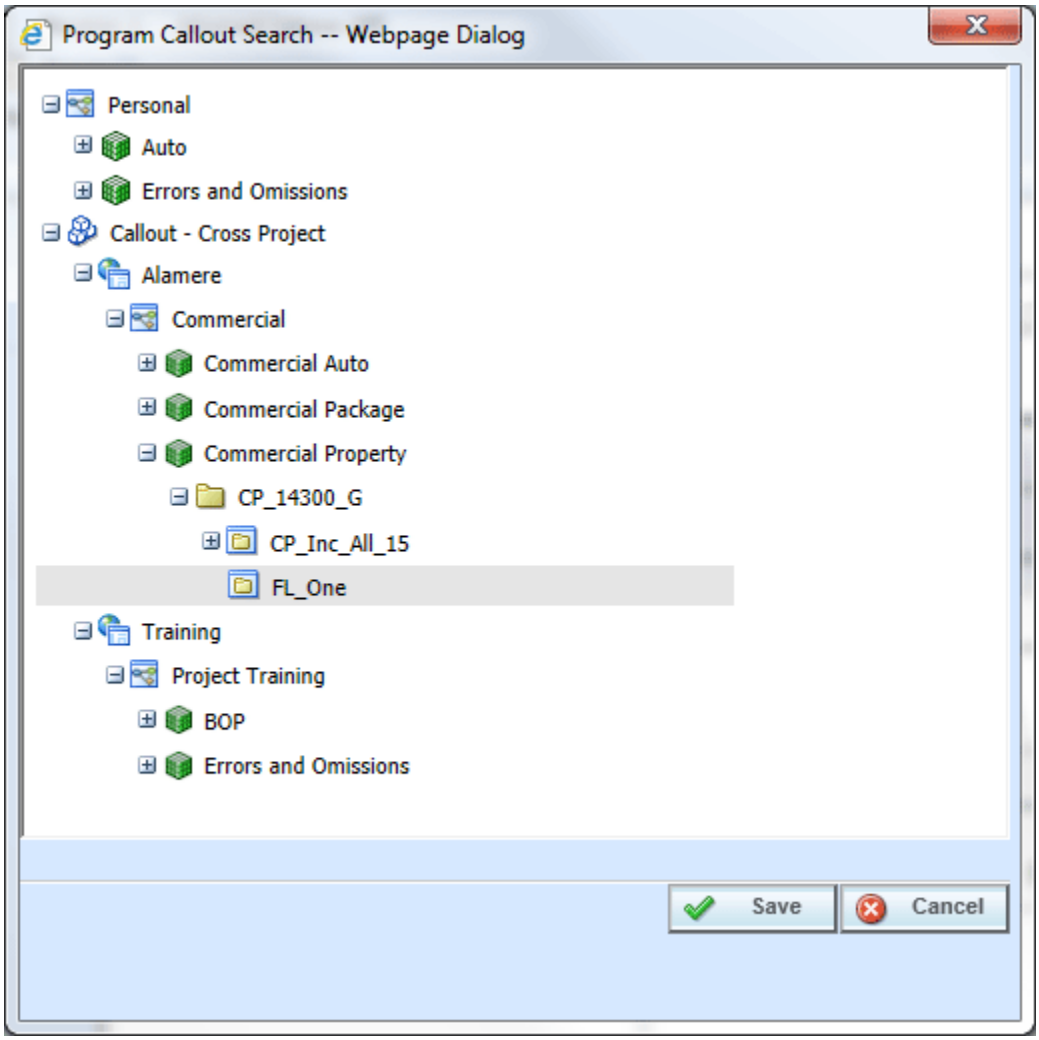


Figure 452 Selecting a Program

4. Select the callout you want.

NOTE: *Programs in projects are listed if a cross project relationship has been established.*

5. The callout is listed as the next step.

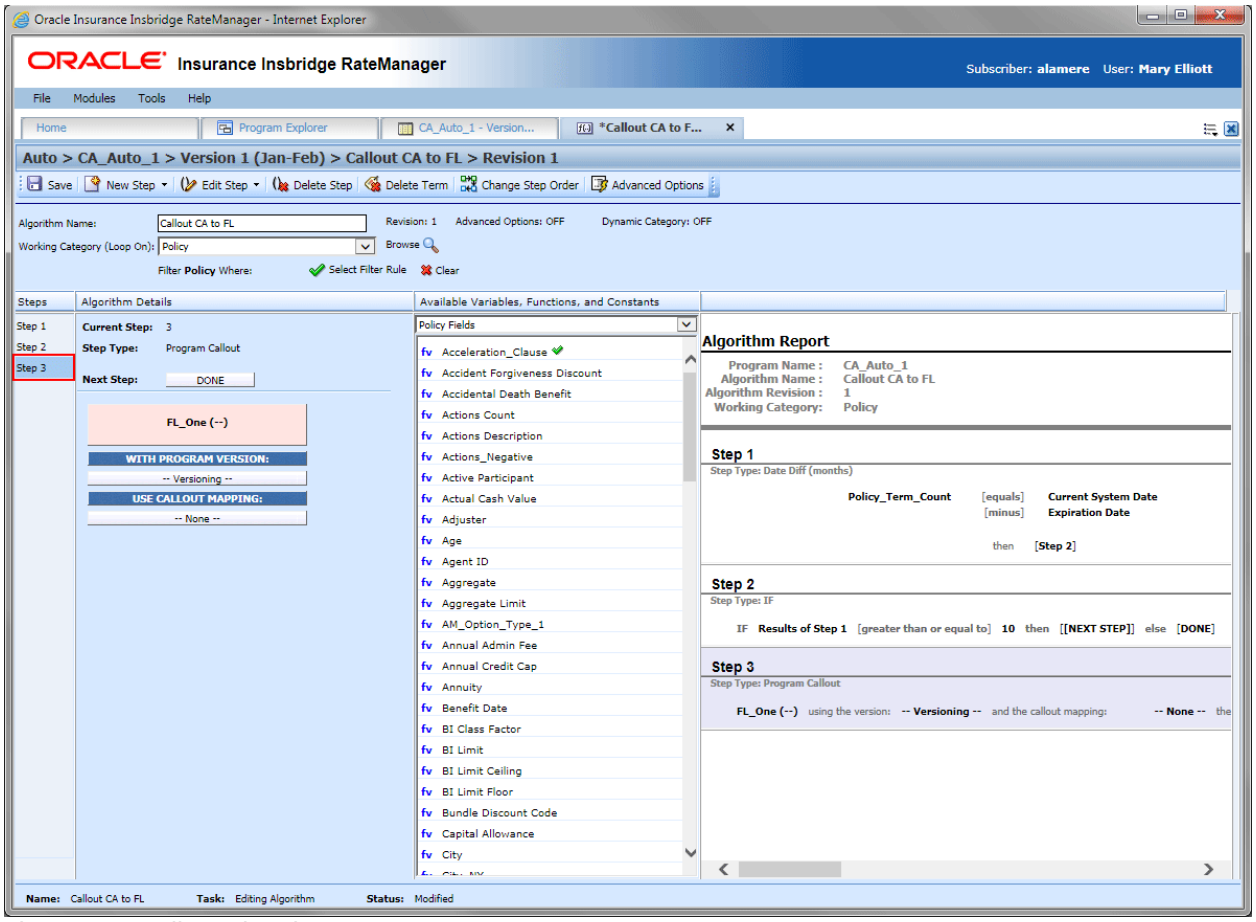


Figure 453 Callout Listed as a Step

6. Save your work.
7. If you have a mapping that you want to use, select Program Callout Mappings from the Available Variables, Functions, and Constants.
8. Available mappings are listed. Select the mapping you need.
9. If you have a program version that you want to use, select the variable that you need.
10. Complete your algorithm and Save.

USING P2L CALLOUTS

P2L callouts are used in steps in local algorithms or driver assignments. To use a callout, select the function – Library Callouts from the Available Variables, Functions, and Constants. The available SoftLibraries will be listed on a separate window. Select the callout that you want to run. This populates the step in the algorithm or driver assignment. For SoftLibraries you also must create a mapping prior to using.

If you choose to have multiple programs in an algorithm, please be aware that a program cannot make a callout back to the initiating program. This is circular logic and will result in an error.

For callouts, step types will not matter. The entire step will consist of the callout. No other entry can be made. Callouts are program specific and are available in local algorithms and driver assignments only.

Prior to using a P2L callout you should have ready:

- Mapping Exceptions – All SoftLibraries require mapping exceptions. Fields and results must be mapped from the calling program to the SoftLibrary and back. No automatic mapping is done on a SoftLibrary.

- SoftLibraries – A SoftLibrary is a specially developed program that performs a specific task. SoftLibraries are loaded into SoftRater and activated in IBFA. Permissions may be required to load and activate a SoftLibrary. Please contact your system administrator

CREATING A P2L EXCEPTION MAPPING

Prior to creating the algorithm or DA, you need to create an association between the **program version** and the SoftLibrary that you want to call. If you do not have a callout mapping for the specified version, nothing is populated when you select Library Callouts from the Available Variables, Functions, and Constants. If the selection area is blank, there's no association. You need to save your work and then create a callout mapping.

You can have multiple P2L mappings for the same library and/or the same program version.

1. Navigate to the domain type→domain→project→folder→program where you want to map a callout.

2. Right click it and select **Map Callouts** from the menu.

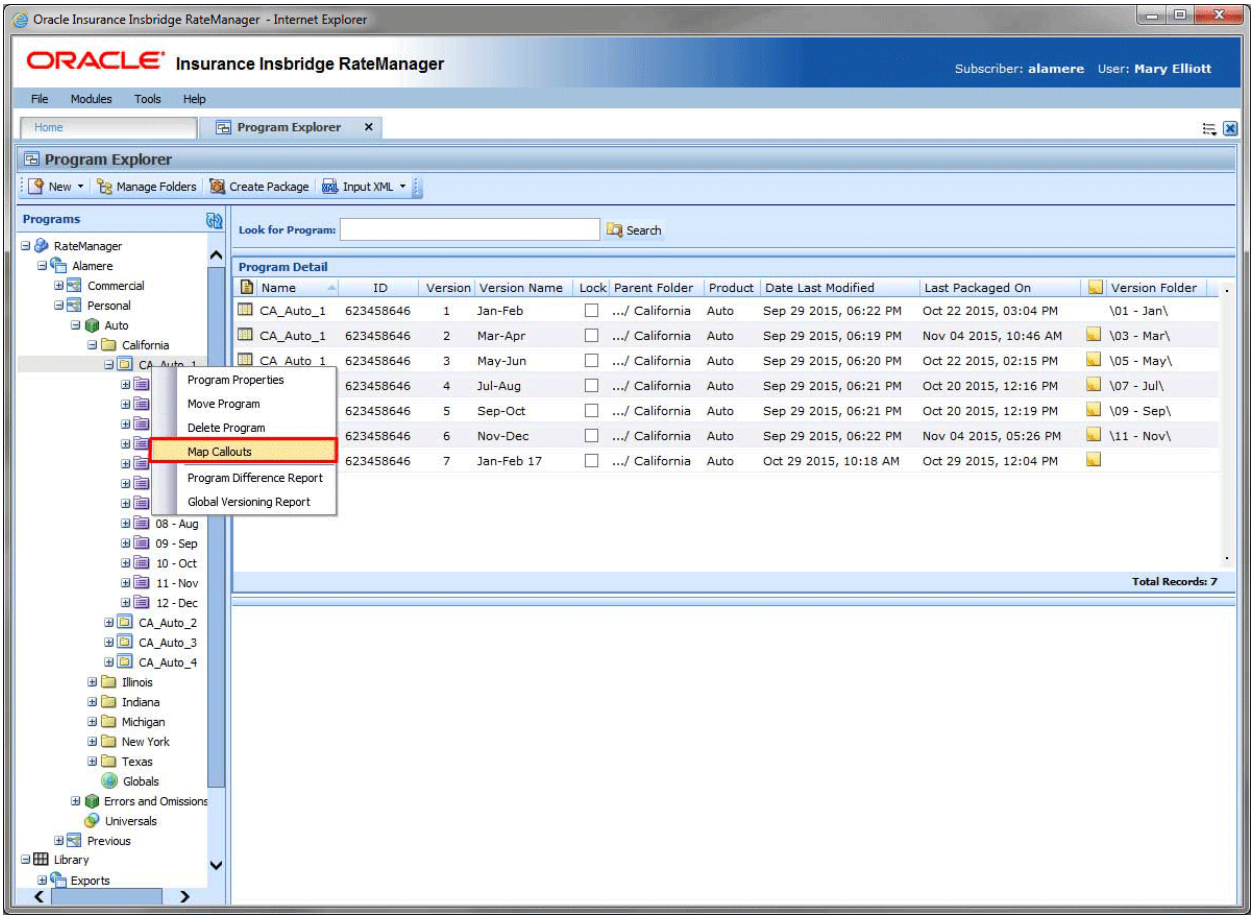


Figure 454 Navigating to Callouts

3. The Map Callouts screen is displayed.

MAP P2L CALLOUTS

At this time, the driver/vehicle category found in Drivers Assignments cannot be mapped.

For mapping exceptions, you must set the Field and Output mappings for each program. SoftLibraries require mappings.

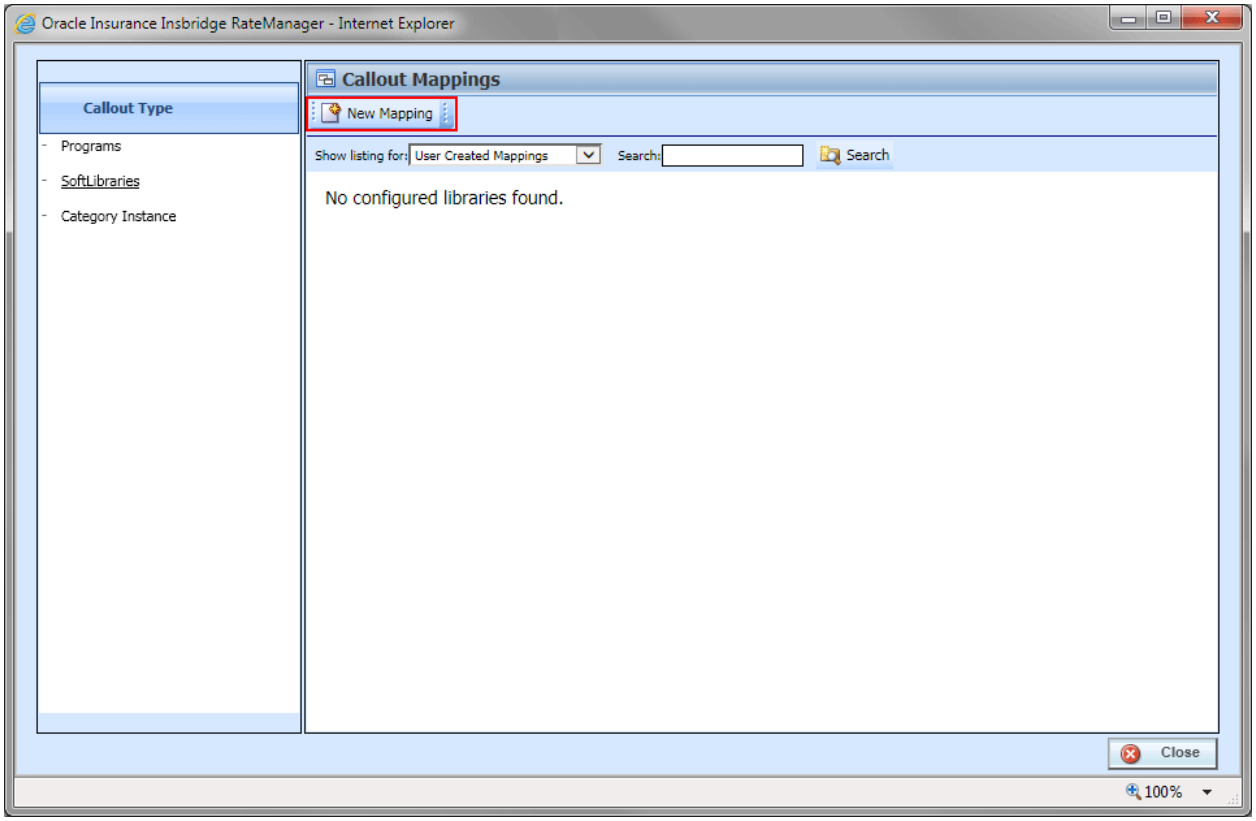


Figure 455 Map Callout Exceptions

Creating a New Mapping

1. Select SoftLibraries as the Callout Type.
2. Click the New Mapping button. The library selection screen is displayed.

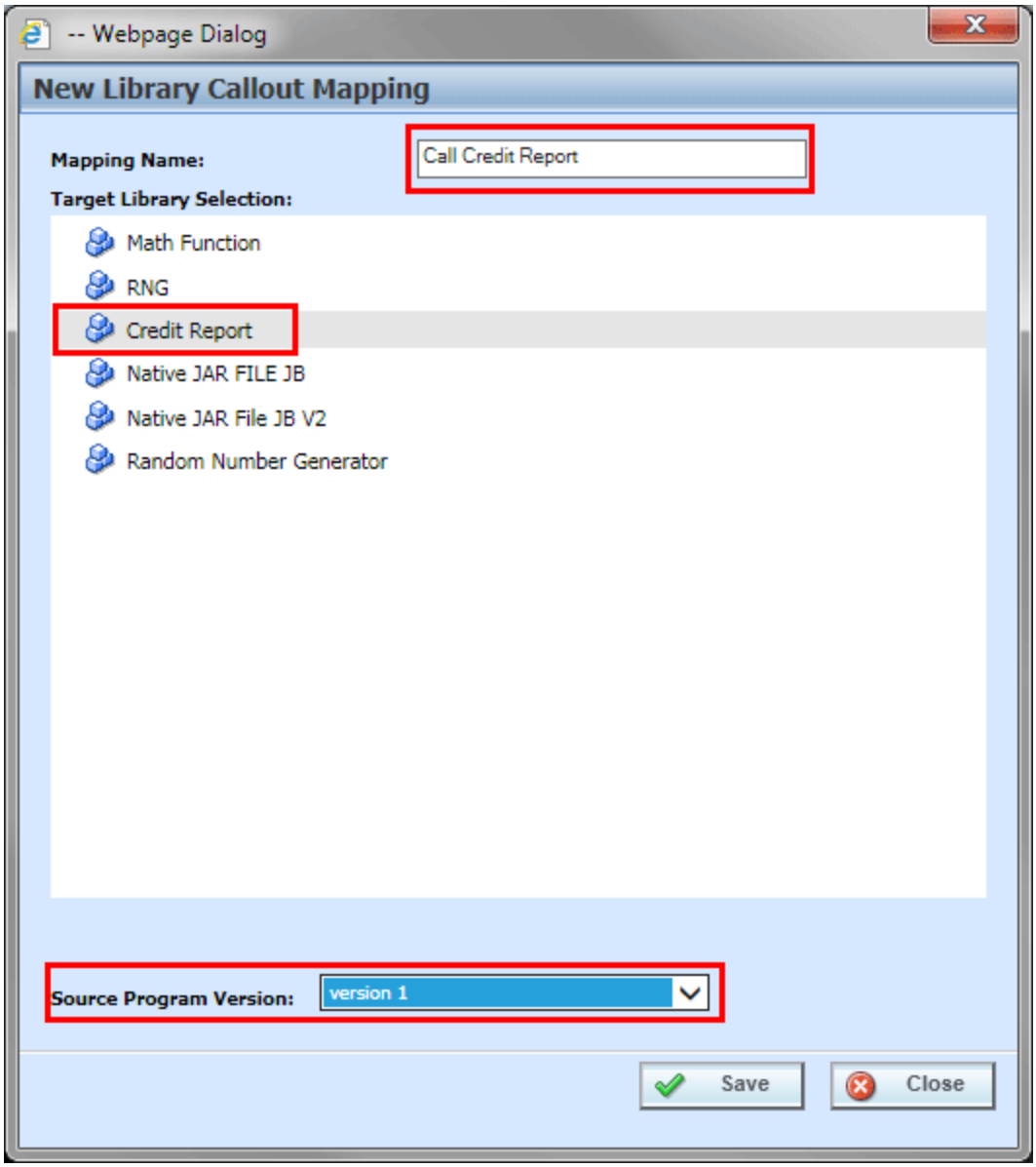


Figure 456 Creating a Library Callout Mapping

3. Enter a name for the mapping.
4. Select the library you want to map to.
5. Click search and select an output mapping group to use.
6. Click Save. The new mapping is listed on the Map Callout screen.

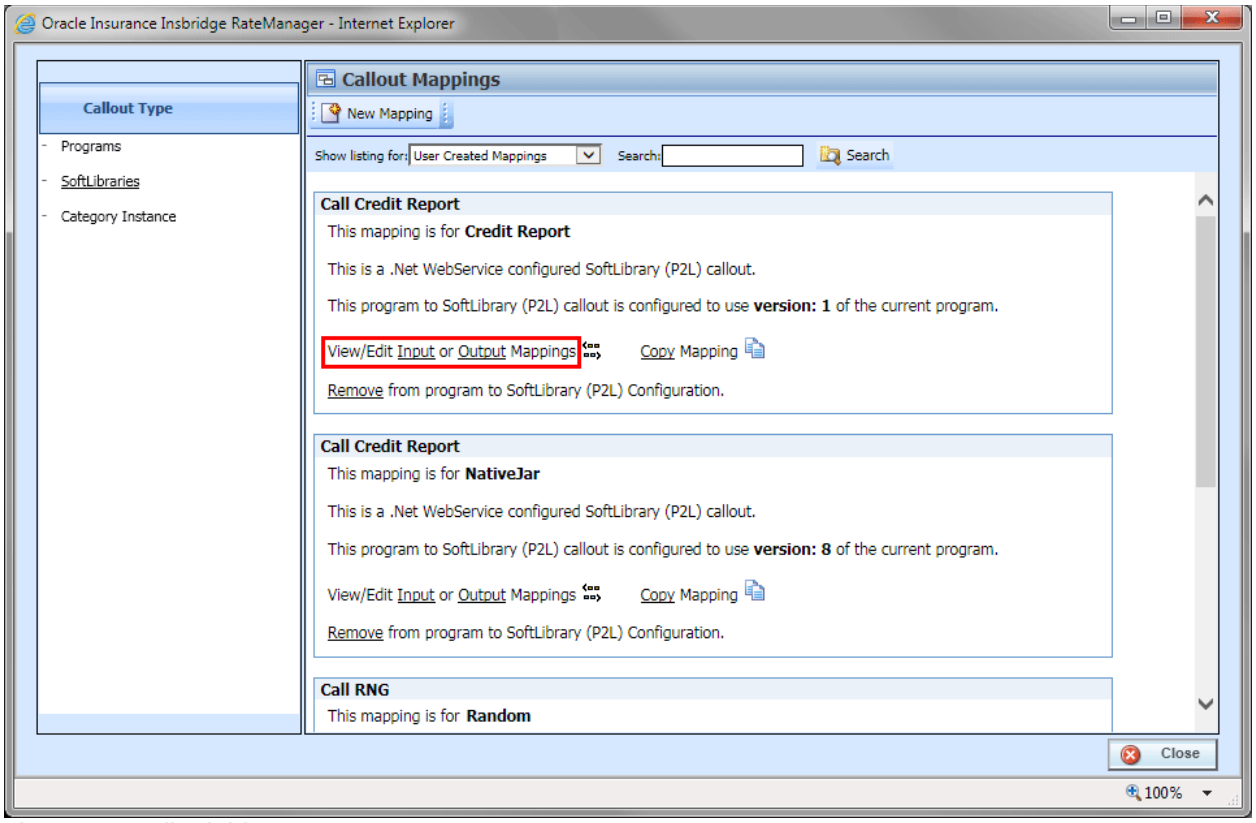


Figure 457 Edit Fields

7. To view or edit exceptions, click the Input link or the Output Mapping link. This opens the exception screen.

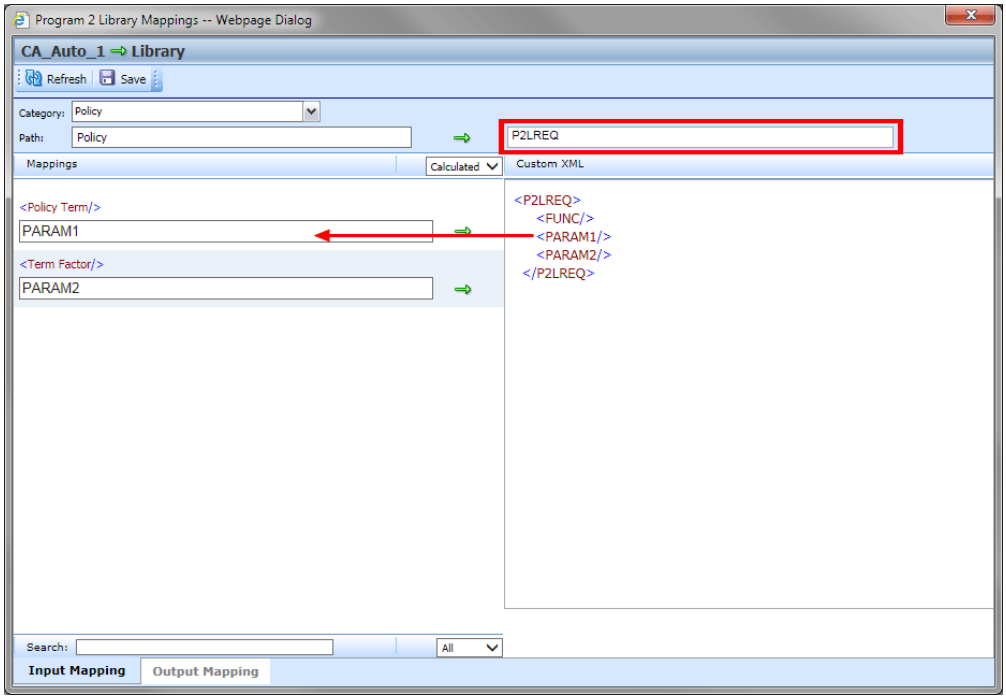
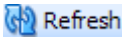


Figure 458 Adding Exceptions

1. Highlight the corresponding category in the library Custom XML field. Drag the category into the path field.
2. Enter in the mappings. Highlight the corresponding mapping field in the library Custom XML area. Drag the mapping into the mapping field.
3. Enter as much information as you need. If your entries do not populate the screen immediately, you can view your entries by clicking Refresh.
4. When complete, select the **Output Mapping** tab at the bottom of the screen. This will open up the output mapping screen where you can map variable to results that are used.
5. Enter in the output mapping path. Highlight the corresponding category in the library Custom XML field. Drag the category into the path field.
6. Enter in the mappings. Highlight the corresponding mapping field in the library Custom XML area. Drag the mapping into the mapping field.
7. Enter as much information you need. If your entries do not populate the screen immediately, you can view your entries by clicking Refresh.
8. Verify your mappings. If they are correct, click **SAVE** to save your entries.
9. Close the Mappings screen. You will be returned to the Map Callouts screen. Your mappings will be in place.

Functionality

The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform on this tab.

	**Refresh:** Refreshes the screen. You may need to refresh to display all exceptions.
	Save: Saves the mapping.
	Direction To: Shows the direction of the exception mapping. This is for fields.
	Direction From: Shows the direction of the exception mapping. This is for outputs.

Searching on Mapping Outputs

You can search for result variables on the SoftLibrary mapping screens. The search is located on the lower right side of the screen,

Enter in your search criteria in the search window. Select if you want to search all variables or only table variables.

Results are displayed in the left side of the screen.

Selecting Input or Output Tab

You can enter input or output exceptions by selecting the appropriate tab on the bottom of the screen.

To Create a P2L Callout Step

If nothing is populated when you select Library Callouts from the Available Variables, Functions, and Constants, there's no associated SoftLibrary for this program version. You need to save your work and then create a P2L callout mapping.

1. Select the local algorithm or driver assignment where you want to use a callout.

2. Select Library Callouts from the Available Variables, Functions, and Constants.

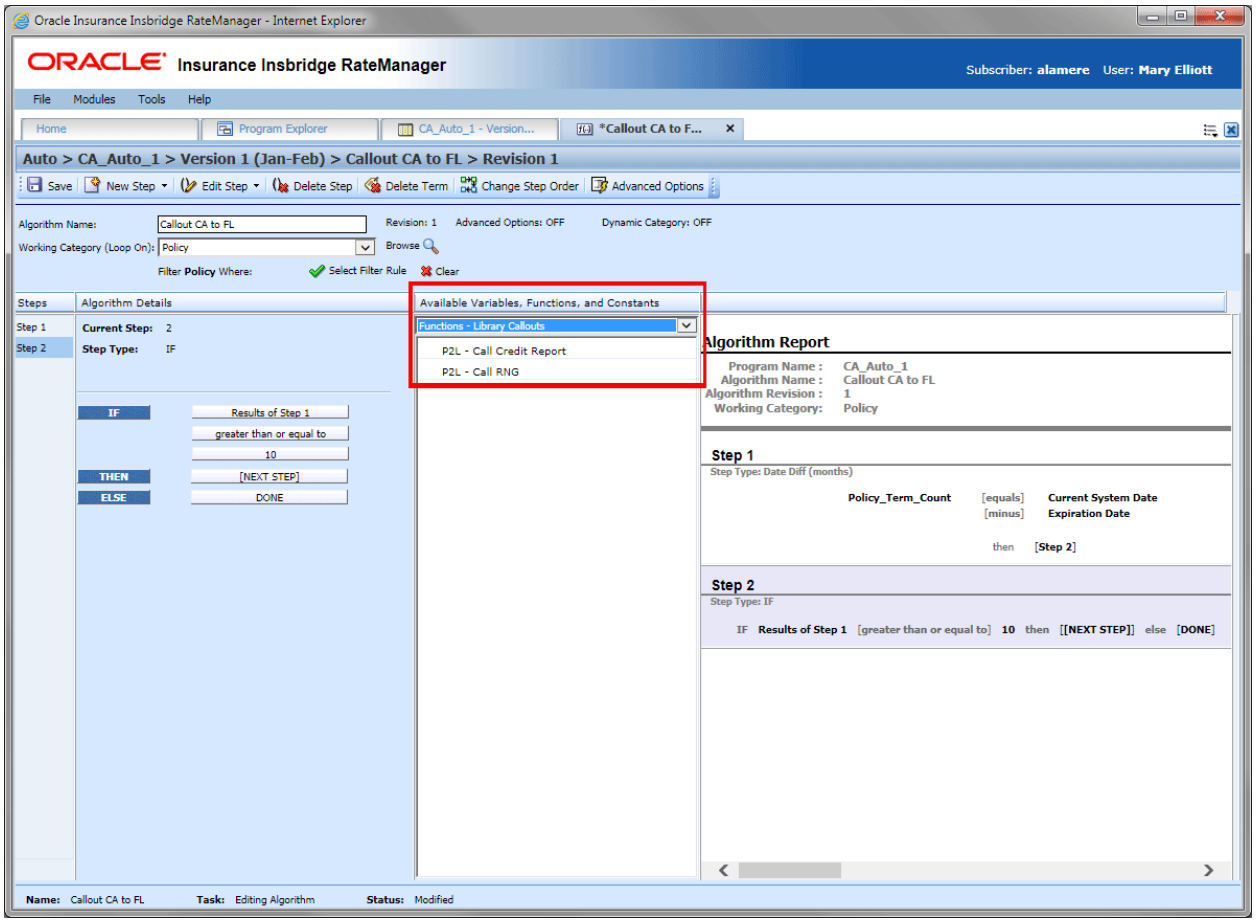


Figure 459 Selecting a P2L Callout

3. Available libraries will be listed.

4. Select the library you want.

5. The library is listed as a step.

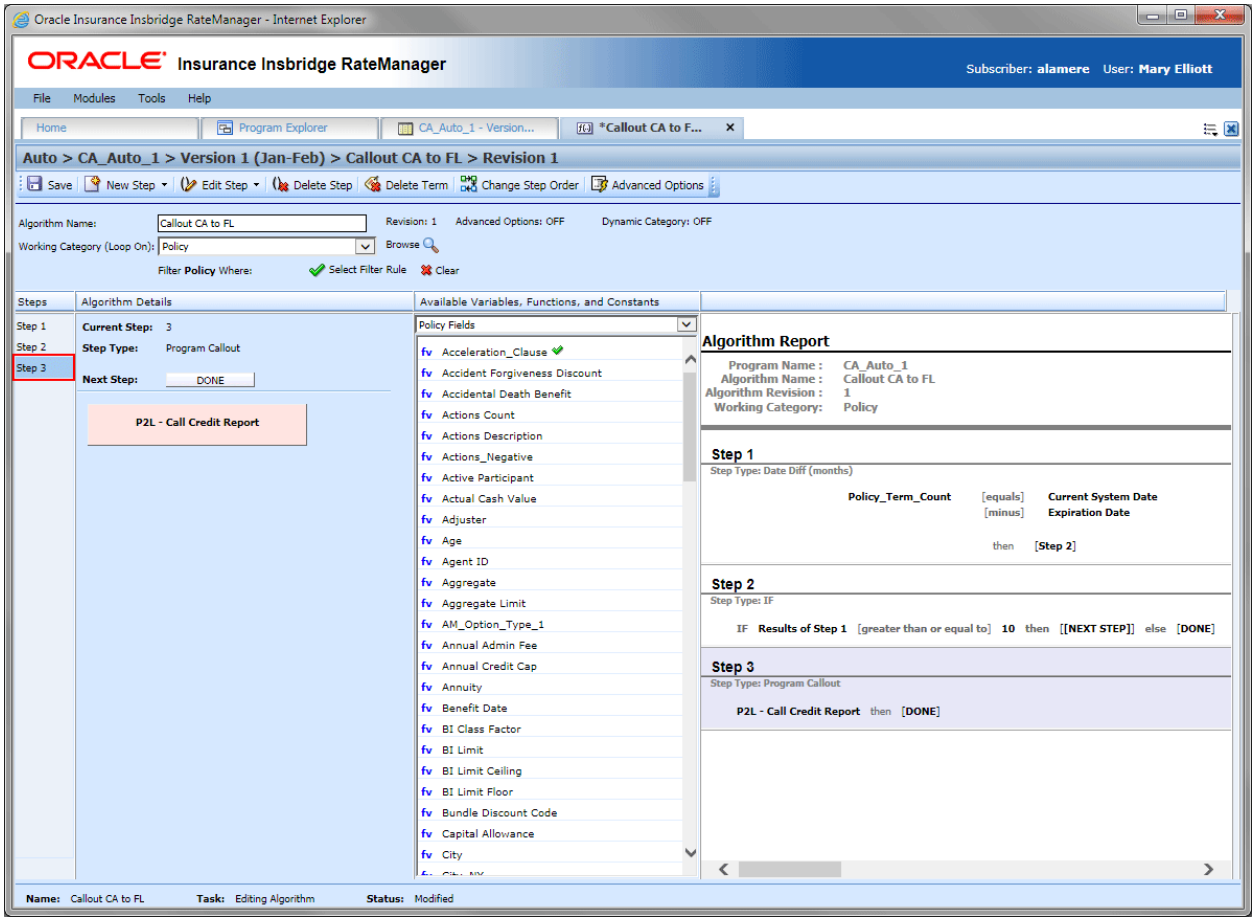


Figure 460 P2L Listed as a Step

6. Save your work.

Removing a Program Mapping

Program exception mappings can be removed from the Map Callouts screen at any time.

1. Navigate to the program where you want to remove the mapping.

2. Right click and select the Map Callouts callouts screen.

3. Select Programs.

4. Select the exception mapping you want to remove.

5. Click the **Remove** link. A warning message will be displayed. Select **OK** to remove the mapping, click **Cancel** to return to the previous screen without deleting the mapping.

A mapping exception that is being used cannot be deleted. If you try to remove a mapping that is being used, an error is displayed. You must remove the mapping from the callout step or delete the callout before removing the mapping exception.

Copying a Mapping

Any mapping can be copied. The mapping is copied exactly. The name is updated. The mapping is in the same program for P2P copies and in the same program version for P2L copies. Mapping exceptions can be altered.

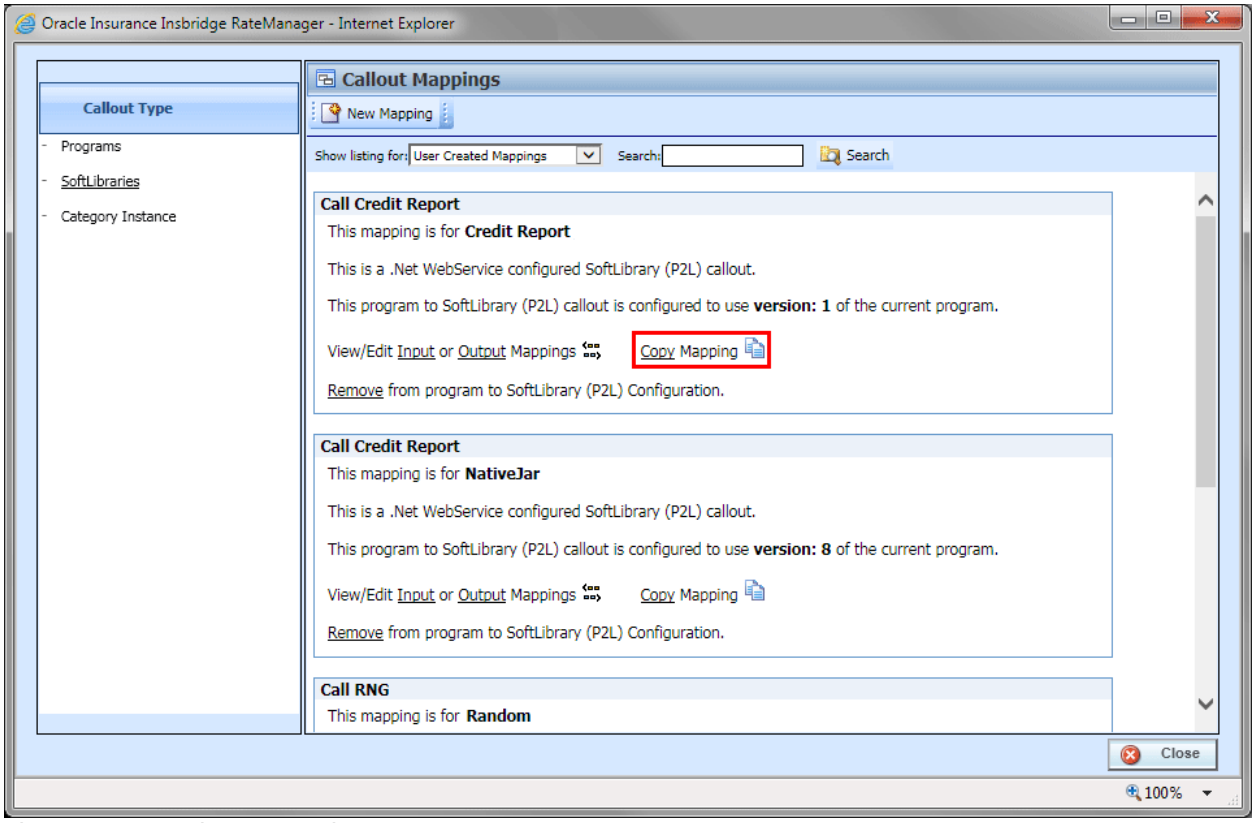


Figure 461 Copying a Mapping

1. Select the mapping where you want to make a copy.
2. Click the Copy Mapping option.

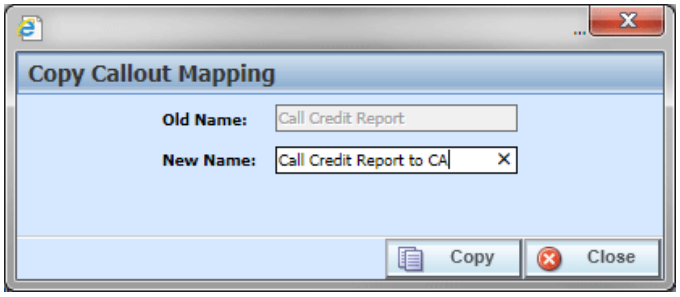


Figure 462 Copy Callout Mapping

3. Enter in a New Name.

4. The mapping will be listed. The mapping is open to exception updates.

Copied or Versioned Programs

Callouts are version specific and function at the version level. When you create a new version or copy a version, the P2P mappings are the same ones as the version where they were created. You may want to adjust these mappings to reflect any changes you are making to the copied or versioned program.

When a change is made to a callout, the change will only be in version where the callout was updated. On copied or versioned programs only, P2P callouts should be verified to make sure the mappings that were brought over are the ones you want.

P2L mappings will not get brought over into the new version. SoftLibraries must be re-registered.

If you enter a copied or versioned program and see an error message on your P2L, you will need to re-register the library.

Implementing Callouts

After creating a callout, you can select it when creating or editing steps for algorithms. Callouts are useful when you have to call an external data provider, such as the motor vehicle office, replacement costs calculators or credit reports. For example, you can place a callout in an algorithm so that the call is placed only after certain criteria have been met. This assures that the call is made against a risk that is more likely to qualify instead of a risk that may not meet the requirements.

You can also use a callout to access an internal database or internal process. Some insurers have central repository of data or applications that are shared company wide. Instead of loading this information into RateManager, a callout can be made from RateManager to obtain the information.

Callouts are listed under Functions: Program Callouts or Functions: Library Callouts. Choose the type of callout you want. The available programs or SoftLibraries are populated underneath. Select the callout you want.

For callouts, step types will not matter. The entire step will consist of the callout. No other entry can be made. Callout selections are available under the project where they were created.

WORKING WITH REVISIONS

Element revisions can be created in both standard programs and in template generated programs, unless locked, at any time. Only unlocked elements can have new revisions added. In standard program revisions, all elements are open for editing. If a variable is locked, you can copy it and then make a new revision from the copy.

When you create a new element revision, it gets created in all versions of the program. Universal, global and local elements are created, edited, copied, and deleted in the same way. Universal, global and local elements are also revisioned in the same way.

Fields, result variables, and constants do not have revisions.

Creating an Element Revision

The new revision is created as the next available revision number. The program does not use the new revision until it is set as the active revision. For table variables you will get a confirmation box asking you if you would like to include any data found. You have to select yes or no to continue.

To Create an Element Revision

1. Navigate to the Program Details tab for the program version that contains the element where you want to create a new revision.
2. Select the element you would like and right click. Select Show/Hide Revisions **f**rom the menu.

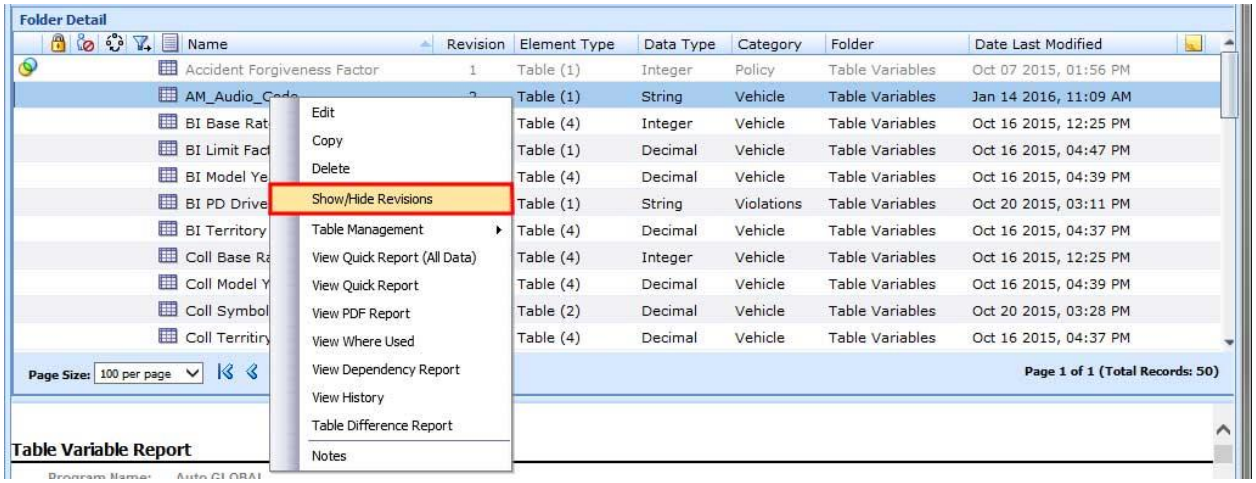


Figure 463 Selecting a Variable to Revise

3. Select the revision you would like the new revision based on and right click it. Select New Revision from the menu. The new revision contains the same information as the revision you select to base it on. If Change Control has been turned on, you must enter that information first.

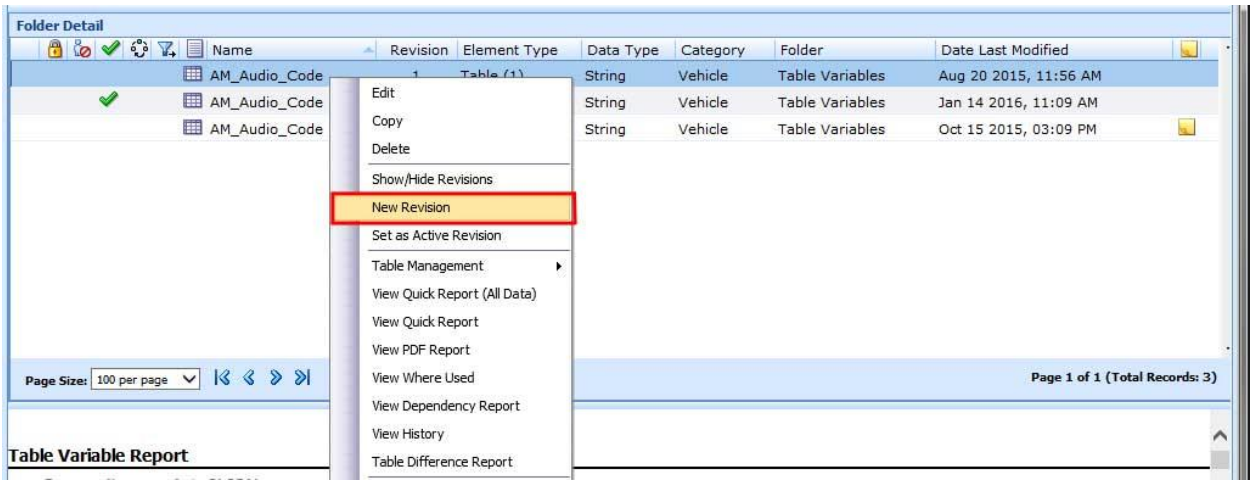


Figure 464 Revising a Variable

4. The new revision is created as the next available revision number and the program details tab listing refreshes to display the new revision. The new revision is open for editing.

You can create as many revisions as you need. You can create a new revision when you want to update an element. By creating a new revision for updates, you leave the last revision intact. This may be helpful if you want to "rollback" to a previous revision or make any comparisons.

New revisions can be created on the universal tab, the global tab or the program details tab. Setting a new revision must be done on the program details tab.

NOTE: *You also can enter notes to any element. Notes may be helpful in keeping track of updates that have been made.*

WARNING: Category Updates to Revisions

Category updates should not be done to a revision. All revisions of a variable should be at the same category level. Changing the category for only some of the revisions of a variable is not supported.

Changing the Active Revision of an Element

Once you have created a new revision, you can set it as the revision to be used during rating. There can be only one active revision in a program version. Changing the active revision in one version of a program does not set it as the active revision for other versions of the program.

The active revision can be changed on both unlocked standard programs and unlocked template generated programs. On a template generated program, if you change the revision from a template generated revision to a revision that you created, the variable will be open for editing. If you change back to the template generated revision, the variable will be locked and no longer open for editing.

The active revision is indicated by a green checkmark.

The default active revision is revision 1. If there is another revision that you want to use, go in to the program details tab and set the revision there.

Active revisions cannot be changed on the universals tab or the globals tab. Active revisions can only be changed on the Program Details tab.

To Change the Active Revision of an Element

1. Navigate to the Program Details tab of the program that contains the element where you want to change the active revision.

2. Select the element you would like to change the active revision for and right click. Select Show/Hide Revisions from the menu.

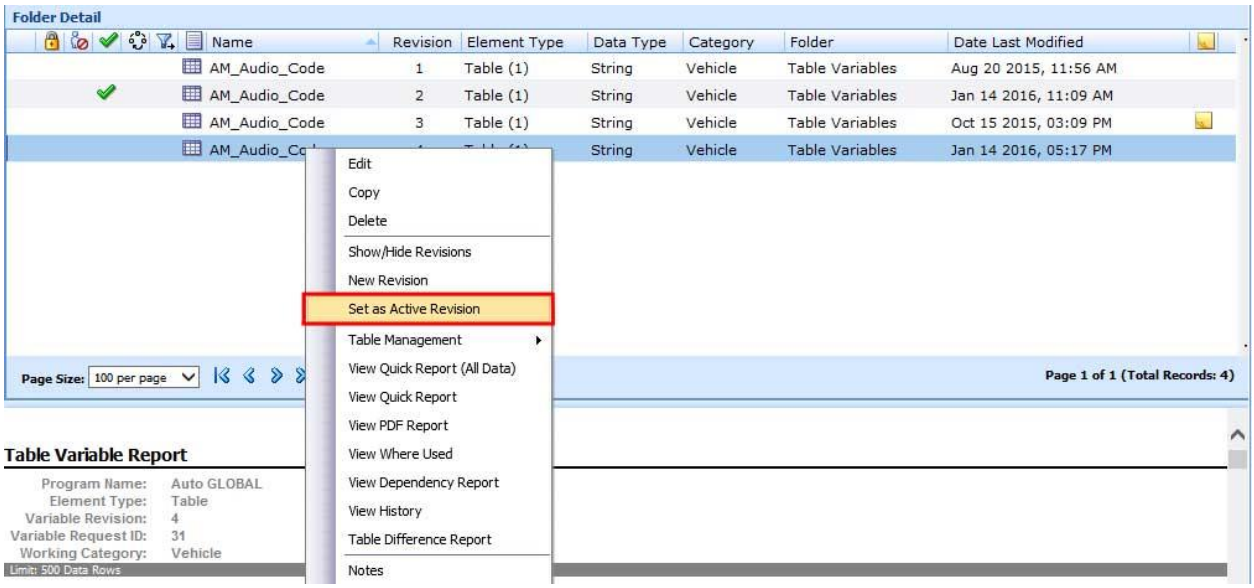


Figure 465 Selecting an Active Revision

3. Select the revision you would like to make the active revision and right click. Select Set as Active Revision.

4. You must confirm changing the active revision.

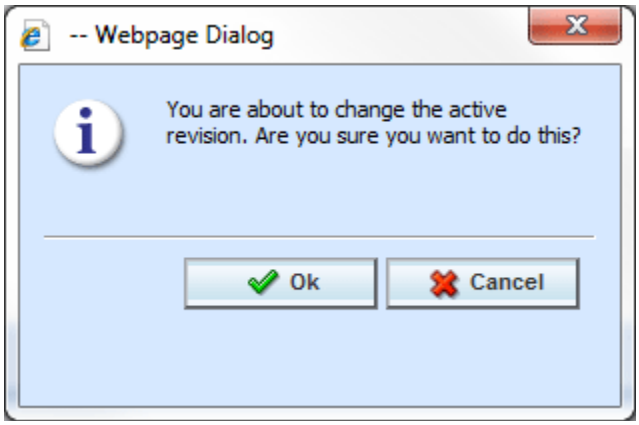


Figure 466 Confirmation Message

5. Select Ok to change the active revision and refresh the tab. Select Cancel to return to the Program Details tab.

6. On the Program Details tab, the new revision has a green checkmark in front, indicating that this revision is the active revision.

Deleting an Element Revision

When a revision is no longer needed, it can either be left in RateManager or deleted. Linked variables, active revisions and template generated revisions cannot be deleted. When you delete a revision, it gets deleted from all versions of the program. If you are uncertain as to whether the revision will be used again, you can leave it in. This saves time if the revision is needed and does not slow down rating.

To Delete an Element Revision

1. Navigate to the Program Details tab of the program that contains element where you want to delete a revision.

2. Select the element that contains the revision you would like to delete and right click. Select Show/Hide Revisions from the menu.

3. Select the revision you would like to delete and right click. Select Delete from the menu.

4. You must confirm the deletion of the element revision.

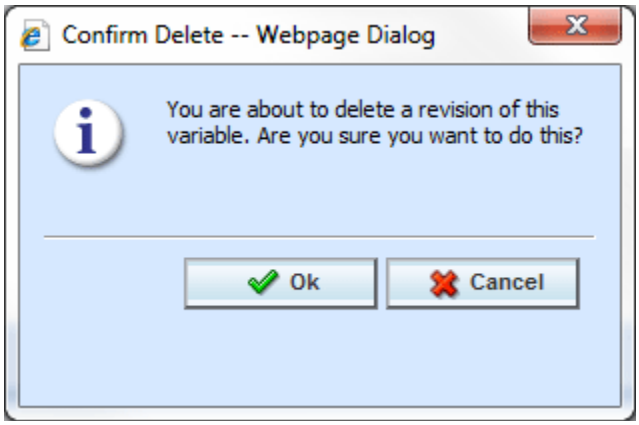


Figure 467 Deleting a Revision

5. Select Ok to delete the element revision and refresh the program details tab listing. Select Cancel to return to the Program Details tab.

If you try to delete the activated revision for the current program version, you will receive an error message instructing you to activate another revision and try again.

If you try to delete a linked variable, you will receive an error message informing you that linked variables cannot be deleted. You can delete the entire variable if necessary.

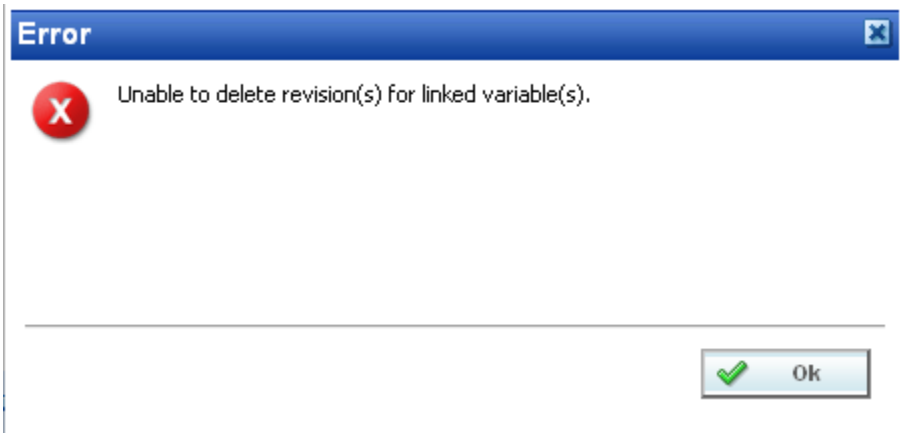


Figure 468 Unable to Delete Revision Due to Linked Variable

Individual element revisions can be deleted from the universals tab and the global tab as well as the Program Details tab.

SEQUENCING

Sequencing activates and orders algorithms and underwriting rules. After creating your algorithms, you set the order in which they should run. For example, if you had a Total Premium algorithm that was calculated from the sum of BI Premium, PD Premium, MP Premium, Comp Premium, and Coll Premium, you would want the Total Premium algorithm to run after all the other premiums had been calculated.

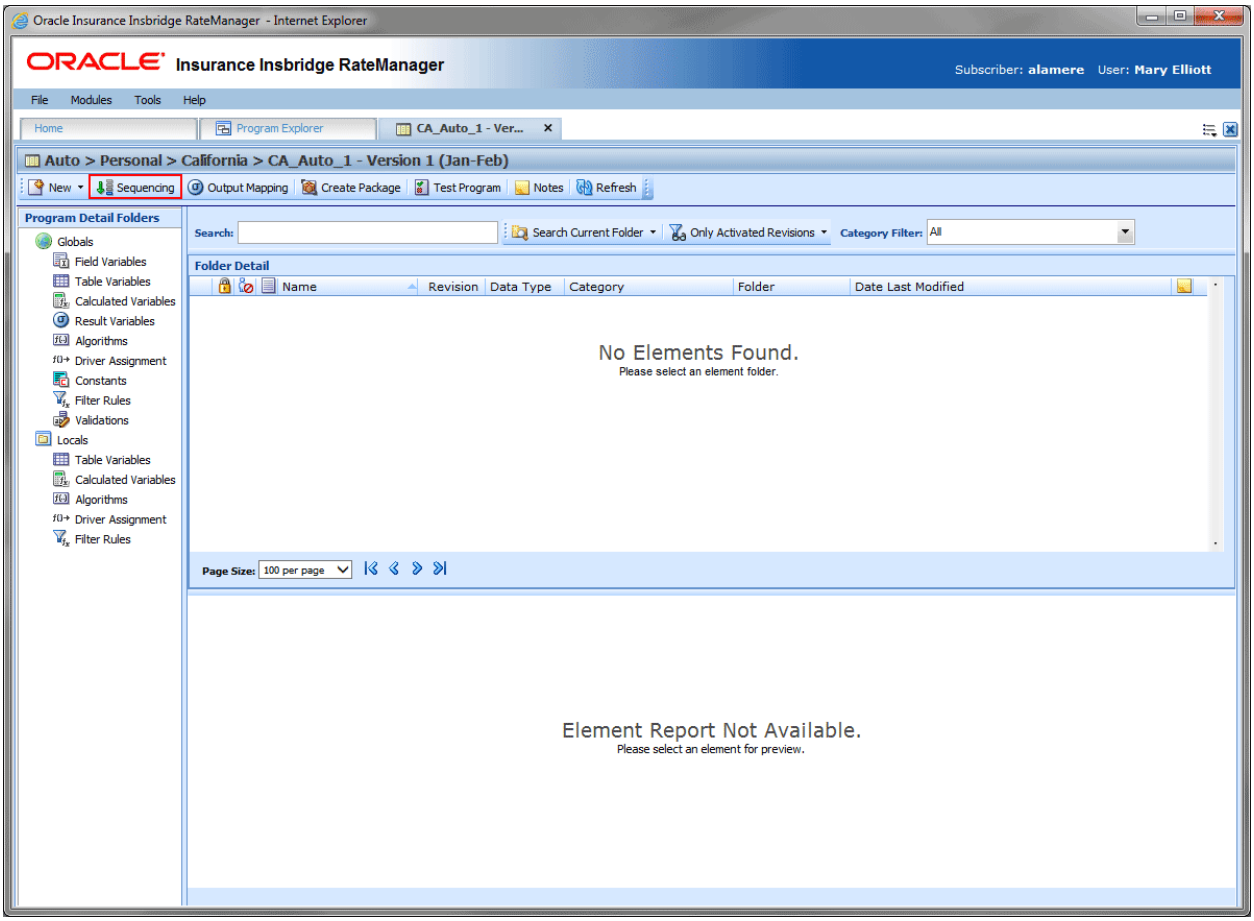


Figure 469 Navigating to Sequencing Tab

Navigating to Sequencing:

Sequencing is not an option on the Program Explorer tab. Sequencing is program version specific and available only on the Program Details tab. You must select the →Folder→Program→Program Version where you want to view or manage sequences. This opens the Program Details tab where you can access the Sequencing tab.

1. Modules→Program Explorer to open the Program Explorer tab.

2. Select the domain type→domain→project→folder→program→program version. This opens the Program Details tab.

3. **Sequencing** is available from a button located on the menu bar. Click Sequencing to be placed on the Sequencing tab.

SEQUENCING TAB

The sequencing tab contains a listing of algorithms and driver assignment scenarios. The default view is for all normal rating algorithms to be listed. To create a sequence, begin on the left side of the tab. Select the type of algorithm you want from the Show Listing For option. Then select the element you want to run first and click the Add button in the center of the tab. The element is placed on the right side of the tab. Continue in order until you have all the elements you want to run in the order you want them to run.

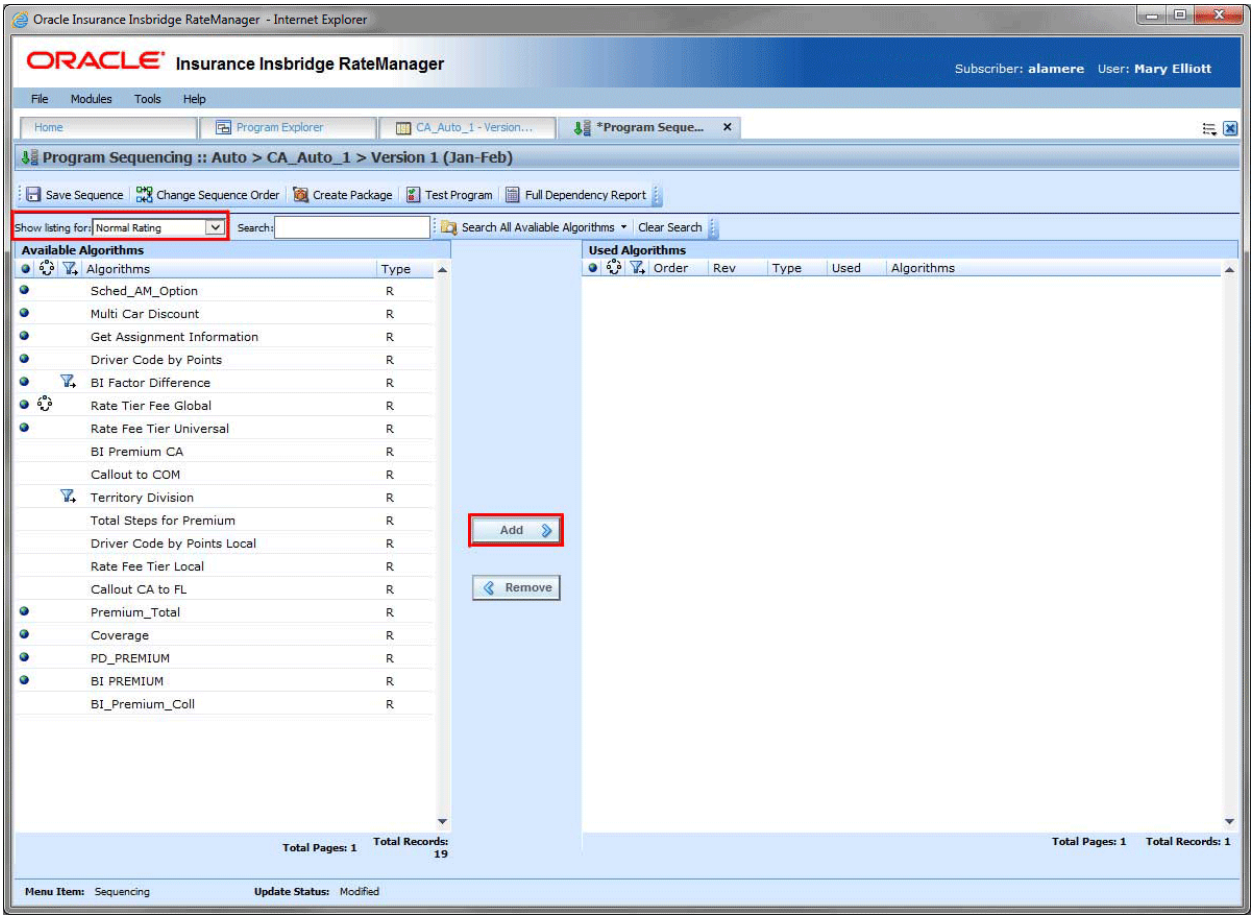
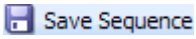
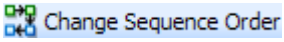
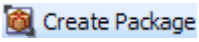
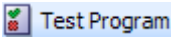
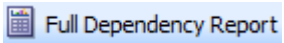
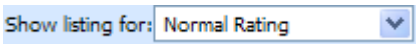


Figure 470 Sequencing Tab

- A globe icon is used to indicate a global element. If there is no icon in front, this element has been created at the local program level.
- An advanced option icon is used to indicate that an advanced option is contained in the element.
- A filter rule icon is used to indicate a filter rule has been applied to the element.
- Driver Assignments are listed in RED.

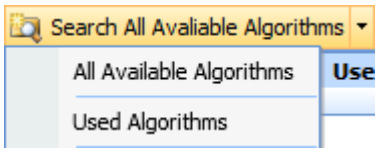
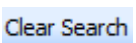
NOTE: *For an auto program, any algorithm that is sequenced before the red colored driver assignment scenario will run before drivers are assigned to vehicles.*

SEQUENCING TAB FUNCTIONS

	**Save Sequence:** Saves the Sequence. If you make a mistake while changing the sequence, you can select **Sequencing** from the menu tree in the left navigation pane to reload the last saved sequence.
	**Change Sequence Order:** Changes the order of the sequences without using the drop and drag feature.
	**Create Package:** Packages the program for rating and testing. See Introduction to Packaging for more information.
	**Test Program:** Opens the Test Program, where you can test and debug a program within RateManager. See Introduction to the Test Program for more information.
	**Full Dependency Report:** Opens the dependency
	**Show Listing For:** Filters the list of available elements to show one of the following: • All • Normal Rating • Underwriting Rule • DA Scenarios - on the Auto product only

Searching Sequencing

When there is a long listing of algorithms, it can be difficult to see if you have already used an algorithm or if an algorithm is available for use. You can search for an algorithm by the selection status, either available or used.

	Search: Enter the criteria that you want to search on.
	**Search Options:** Select what you want to search. There are two options: • **All Available Algorithms:** These are the unselected algorithms on the left side of the tab. Only the algorithms displayed are searched. • **Used Algorithms:** These are the selected algorithms on the right side of the tab.
	**Clear Search:** When you are done searching, you can clear the search field and restore the screen to the previous view.

The elements that are included in the available search are determined by the Show Listing for selection. If only normal rating algorithms are listed, then only normal rating algorithms will be searched. No Underwriting or Driver Assignments will be included. If you want to search All algorithms, make sure the Show Listing For selection is All.

Navigating the Sequence Tab

The Sequence tab has navigation features:

- **Column Sorting:** You can sort algorithms by order number or algorithm column headers. The default view is numeric (lowest to highest) by Order number. To sort by a different column header, click the column header you want and the current results are re-sorted. You can sort results as many times as you want.

  Sorting does not filter results. It only rearranges the order in which they are displayed.

- **Paging:** The lower portion of the screen contains information regarding the total number of algorithms available or used and the total number of pages. Page numbers are displayed when you drag the scroll bars.

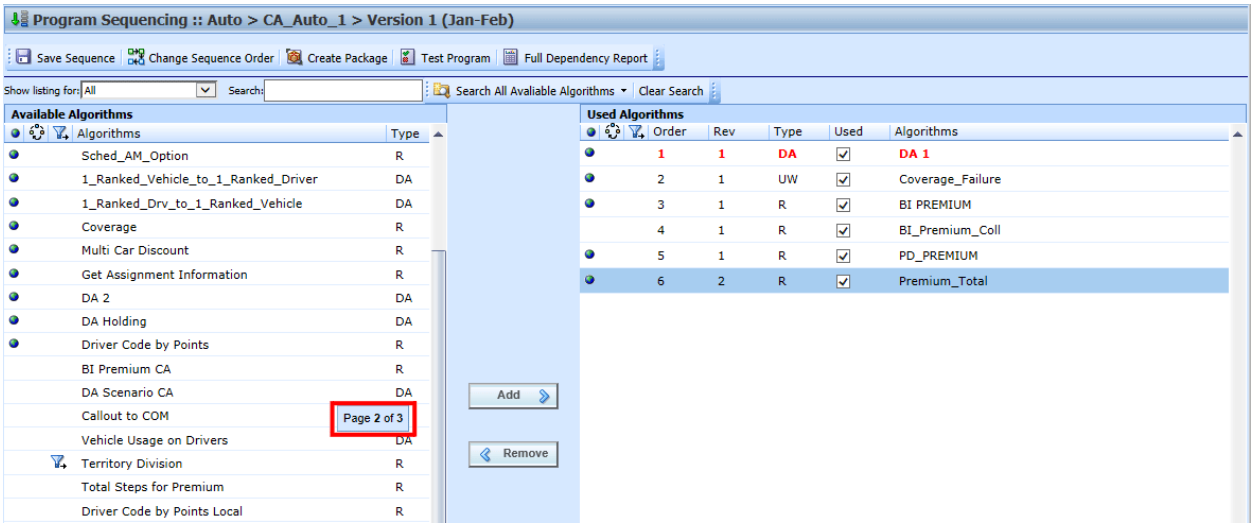


Figure 471 Sequence Paging

- **Reports:** The Used algorithms have two reports available on the right click menu. View Dependency Report and the View Dependency Report All Algorithms Above.

Sequencing Algorithm/Driver Assignment Listings

The program sequencing screen is divided into two areas with two buttons in between.

- **Available Algorithms/Driver Assignments:** The default listing on the left side of the screen shows alphabetically all rating algorithms that have been created for this program version. If you want to view the underwriting rules or driver assignments, select your choice from the Show Listing For selection box. When an algorithm is added from the available side, it will no longer be listed there. It will be listed in the used algorithms side.

- **Used Algorithms/Driver Assignments:** The listing on the right side of the screen shows all algorithms that have been added to the sequencing and the order in which they will be executed. When an algorithm is removed from the used side, it will no longer be listed there. It will be listed in the available algorithms.

- **Add:** Adds the selected algorithm or driver assignment from the list of available algorithms and driver assignments to the sequence. If the algorithm or driver assignment is already in the sequence, an error message will be displayed. Each algorithm and driver assignment can be added once.

- **Remove:** Removes the selected algorithm or driver assignment from the sequence.

NOTE: *You cannot save an empty sequence. All sequences must have at least one algorithm.*

Creating a Sequence

Sequences are program version specific. When a new program version is created, the sequence is inherited from the prior version. Any changes made to the sequence remain in the new version only. The sequence used by the prior version is not affected.

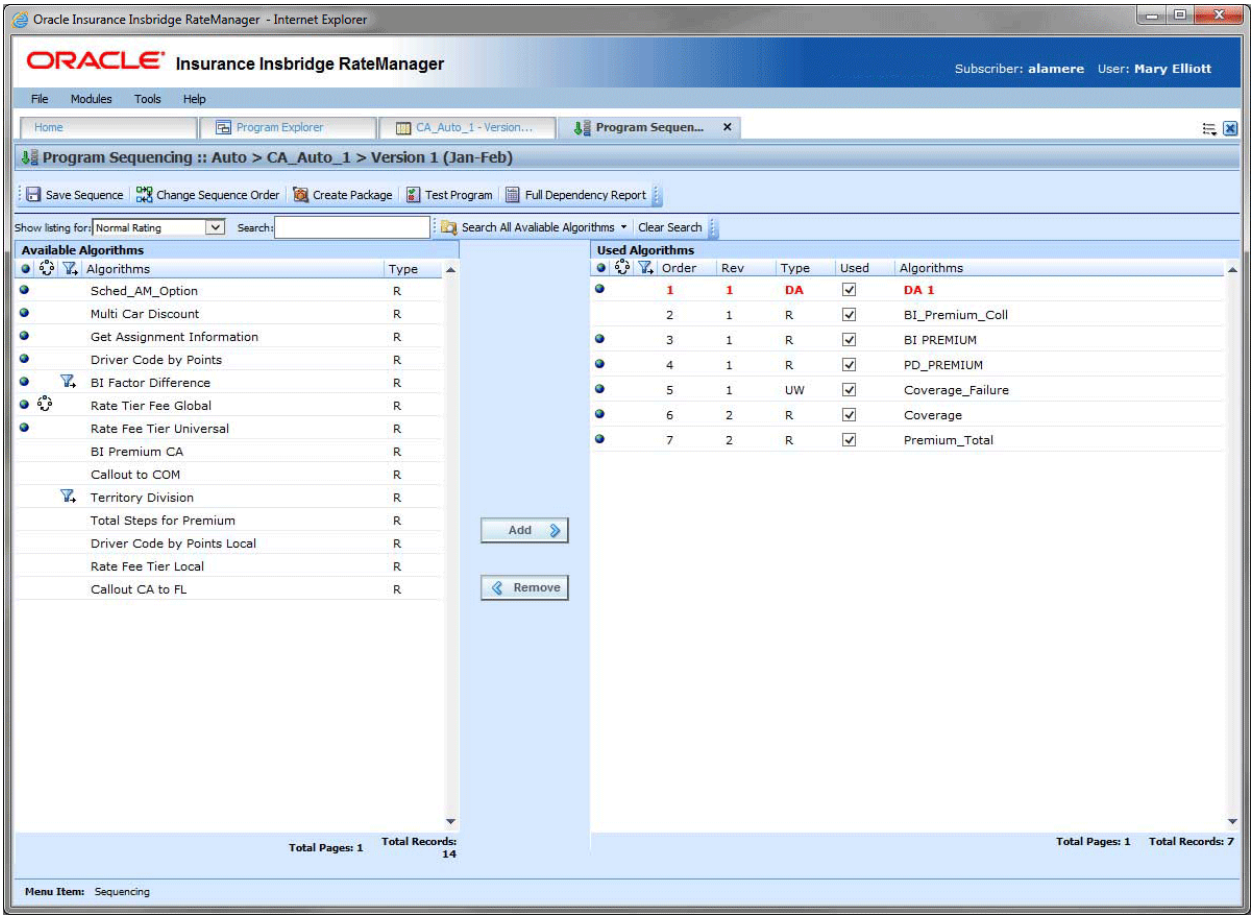


Figure 472 Completed Sequence

Once a sequence has been created, it cannot be deleted, only edited.

1. To add an algorithm or driver assignment to the sequence, you can either highlight it and click **Add** or double-click the algorithm or driver assignment. Add algorithms in the order in which they need to be run.

2. Once the algorithms and driver assignments are activated and sequenced, click **Save Sequence** to save your work.

NOTE: *If you have created an algorithm or driver assignment and it does not appear in the list, be sure the Show Listing For selection box is set to filter correctly.*

An algorithm can be used once per sequence. If you have an algorithm that needs to be run more than one time, you will need to create an algorithm for each time it needs to run.

You can add as many algorithms and driver assignments as you need. The first algorithm you add to the used algorithm side will be the first one executed, the second one added will be the second one executed and so on. You can change the order after you have made a sequence.

All algorithms and driver assignments listed are the active revision for that program version. The revision will be displayed on the left side of the screen in the **Rev** column. If the revision you need is not brought over, you must return to the program and change the active revision. The revision will be updated in the sequence.

All algorithms and driver assignments are active when brought into the sequence column. A checkmark will be in the **Used** column. If you want to leave an algorithm in the sequencing, but do not want it executed, then remove the check next to the algorithm name.

Every algorithm and driver assignment created at the local program level and at the global level will be listed. You may not need some of these algorithms or driver assignments.

At least one field must be used in the program for each category in the sequence. Every algorithm is created with a working category. If you select an algorithm and list it in the sequence but do not have for the working category where that algorithm was created, then program will not package.

NOTE: *If you have made changes to your Sequence, but have not saved, RateManager will prompt you to save before allowing you to navigate to another tab.*

Multiple Selection Add or Remove

Multiple algorithms can be added to the sequence or removed from the sequence by holding the Shift key or Ctrl key when making your selections. The function is the same for the Available Algorithms side or the Used Algorithm side. Multiple selections can be used on any listing screen; Normal Rating, Underwriting Rule, DA Scenario, or All.

To Select a Range of Algorithms

Selecting a range of algorithms can be a fast way to create a sequence but may require moving algorithms to get the exact sequence you want.

The first click you make while holding the Shift key marks the first selection. Keep holding the Shift key and make the second click. This marks the last selection. All selections in between to two are highlighted and will be added or removed.

If the algorithms are being added, the first algorithm selected is listed first, the last algorithm selected is next. The algorithms between the first and last are listed in the reverse order as the Available Algorithm listing.

If you are removing algorithms, the order does not matter and the algorithms are returned to the Available Algorithm listing in alphabetical order.

To Select Algorithms in Sequential Order

Selecting the order of how you want your algorithms to run can be a faster method of creating a sequence as well. The algorithm will show as selected but does not indicate what order it has been selected in.

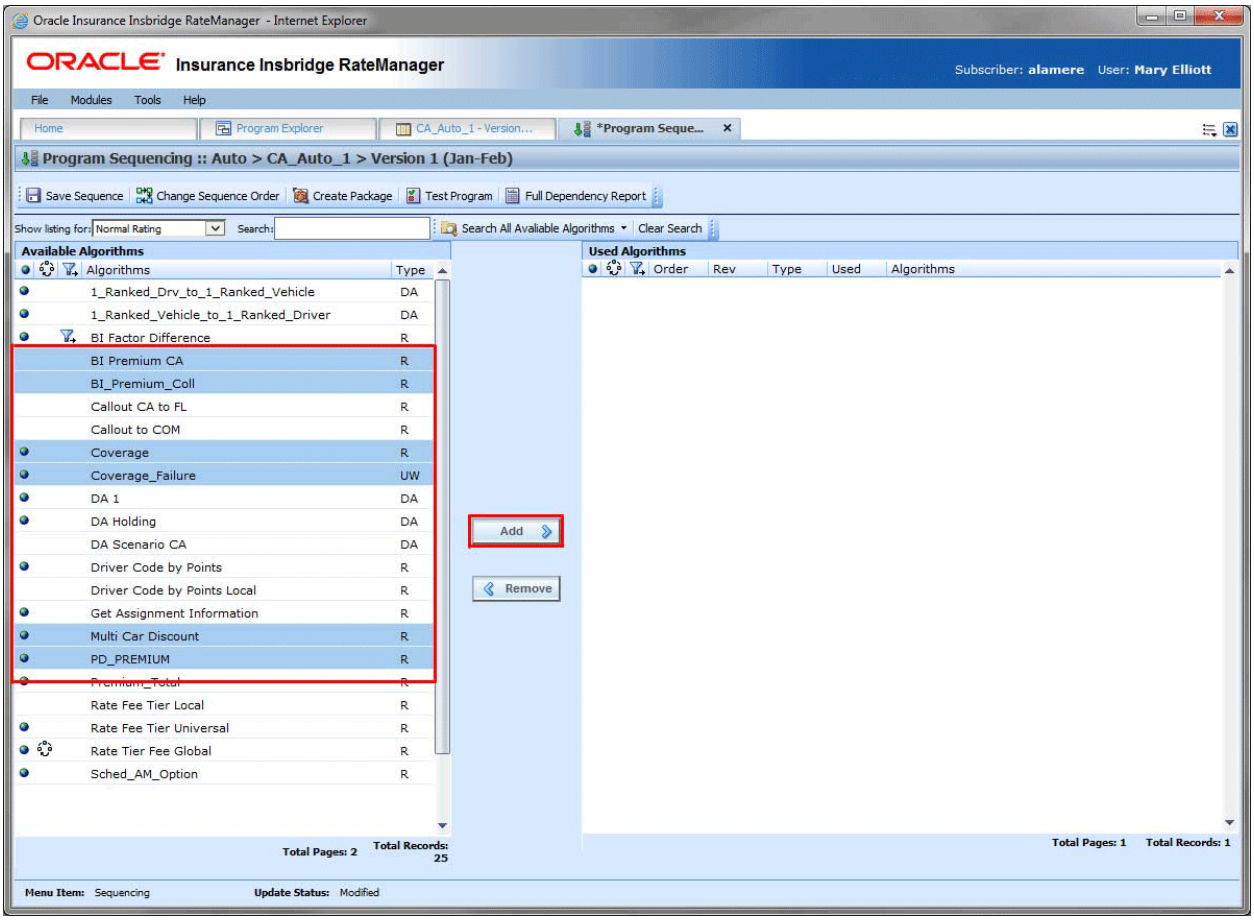


Figure 473 Multiple Selections to Create a Sequence

If you are creating a sequence, hold the Ctrl key down and click the first algorithm you want in your sequence. Click the next algorithm you want to run and continue until you have the sequence you want. Click the Add button to move your sequence over to the Used Algorithms.

If you are removing algorithms, hold the Ctrl key down and make your selections. Click the Remove button to move your sequence over to the Available Algorithms. The order does not matter and the algorithms are returned to the Available Algorithm listing in alphabetical order.

A selection can be removed by holding the Ctrl key and clicking the algorithm a second time. The algorithm is no longer highlighted and no longer included in the sequence.

Changing Sequence Order

Any algorithm or driver assignment can be moved in the sequence order. Sequence order can be changed at any time. If you change sequence order after packaging, you will need to re-package in order for the sequence to be used when rating.

There are two ways to change the execution sequence: drop and drag or click **Change Sequence Order**.

Change Sequence Order Button

1. Select the algorithm or driver assignment you want to move and click **Change Sequence Order**. A separate screen will be displayed.

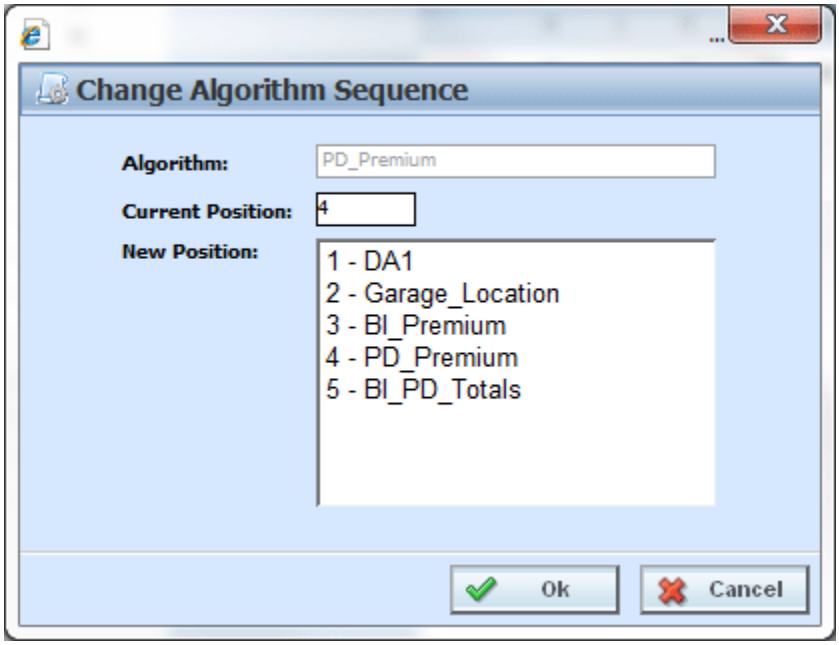


Figure 474 Change Sequence

2. The current position is displayed at the top of the screen. Select where you want the algorithm or driver assignment to be placed.

3. Click **OK**. You will be returned to the program sequencing screen. Your algorithm or driver assignment should be placed where you chose.

Drop and Drag

Use drop and drag when you want to move an algorithm up or down on the current page. If you want to move from one page to another, you must use the Change Sequence Order button. To use the **drop and drag** method, highlight the algorithm or driver assignment you want to move and then select the line *below* where you want to drop the algorithm.

For example, if you want to move an algorithm to the fourth place, then drag the algorithm and hold over the fifth algorithm. When you drop the algorithm, it will be moved into the position above the algorithm where you are currently positioned.

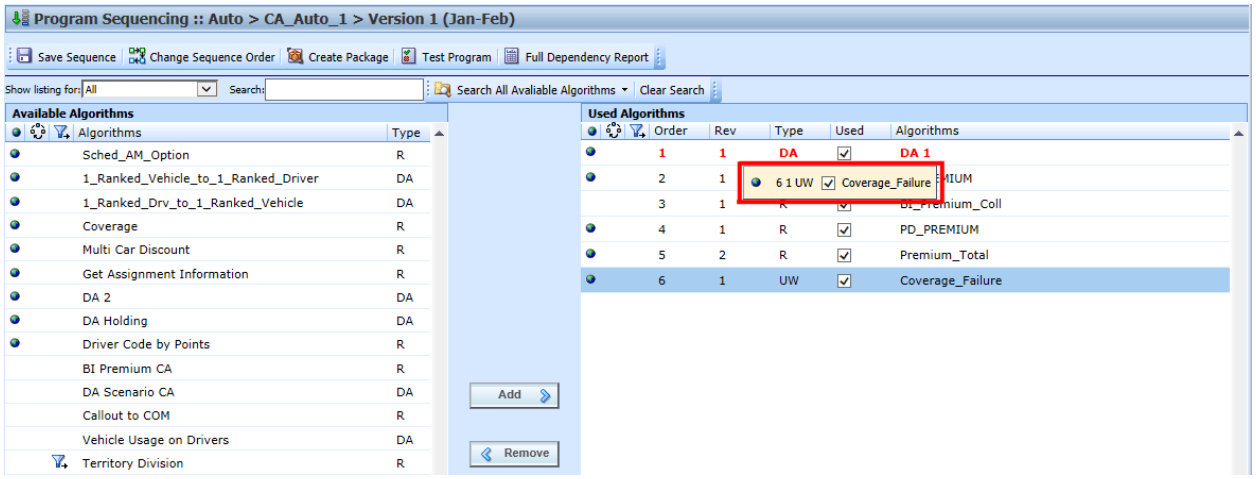


Figure 475 Changing Sequence Order Using Drop and Drag

When you are dragging the algorithm you want to move, it will be encased in a box.

Removing an Element from a Sequence

To remove an algorithm or driver assignment from the sequencing, highlight the element you want to remove and click **Remove**. The element is removed from the used list.

You cannot save a sequence without any algorithms or driver assignments. There must always be at least one algorithm or driver assignment.

WARNING: ***For an auto program, algorithms that use both driver and vehicle information must be sequenced to run after driver assignment***.

OUTPUT MAPPING

The output mapping function is where you can select the results that you want displayed in the output file. You can select any number of elements you want; result, field, table or calculated variables, global or local. Results are mapped to the rating output file based on how you want to view the data. This might include premiums, rating factors, driver ages, etc.

The Output Mapping tab is also where rating worksheets are managed. A Rating Worksheet in RateManager is an XML document that conveys the basic information used to establish individual rates and premiums. The output is based on the current value of the terms at that point in the execution of the step. For example, if result A is being used in step 3 of algorithm 1 and then again in step 5 of algorithm 10, the value of result A will be shown twice in the worksheet – once for algorithm 1 and once for algorithm 10. You select the steps where you want to see the current values. The output reflects the current value at the point of execution of the algorithm. The XML document can be carried into most existing systems where it can be used and formatted in a report to fit your needs.

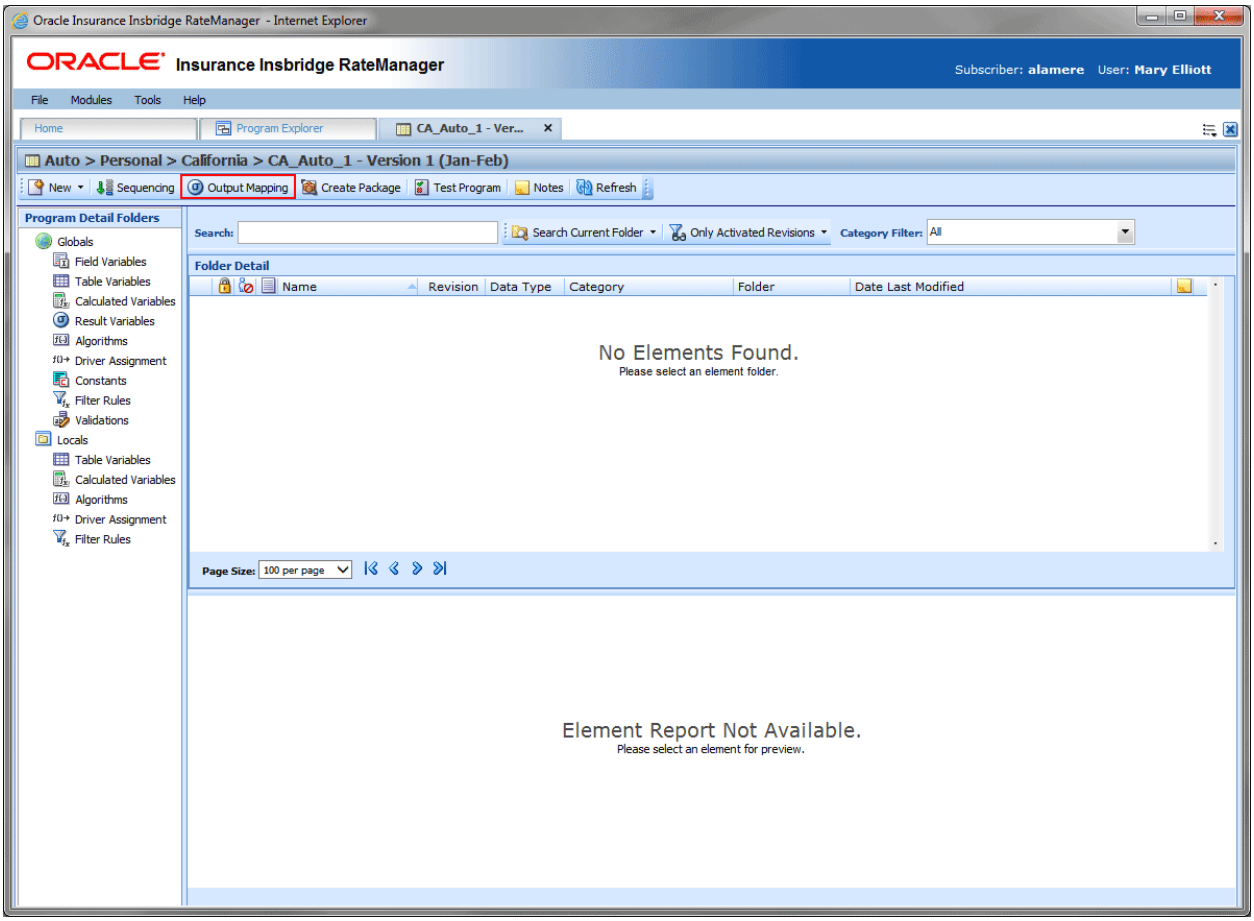


Figure 476 Navigating to Output Mapping Tab

Navigating to Output Mapping

Output mappings and worksheets are program version specific and are not an option on the Program Explorer tab. You must select the Program Version where you want to view or manage output mappings. This opens the Program Details tab where you can access the Output Mapping tab.

1. Select the domain type→domain→project→folder→program→program version. This opens the Program Details tab.

2. **Output Mapping** is available from the top menu bar.

OUTPUT MAPPING TAB

The Output Mapping tab shows all the output groups that have been created for a particular program version.

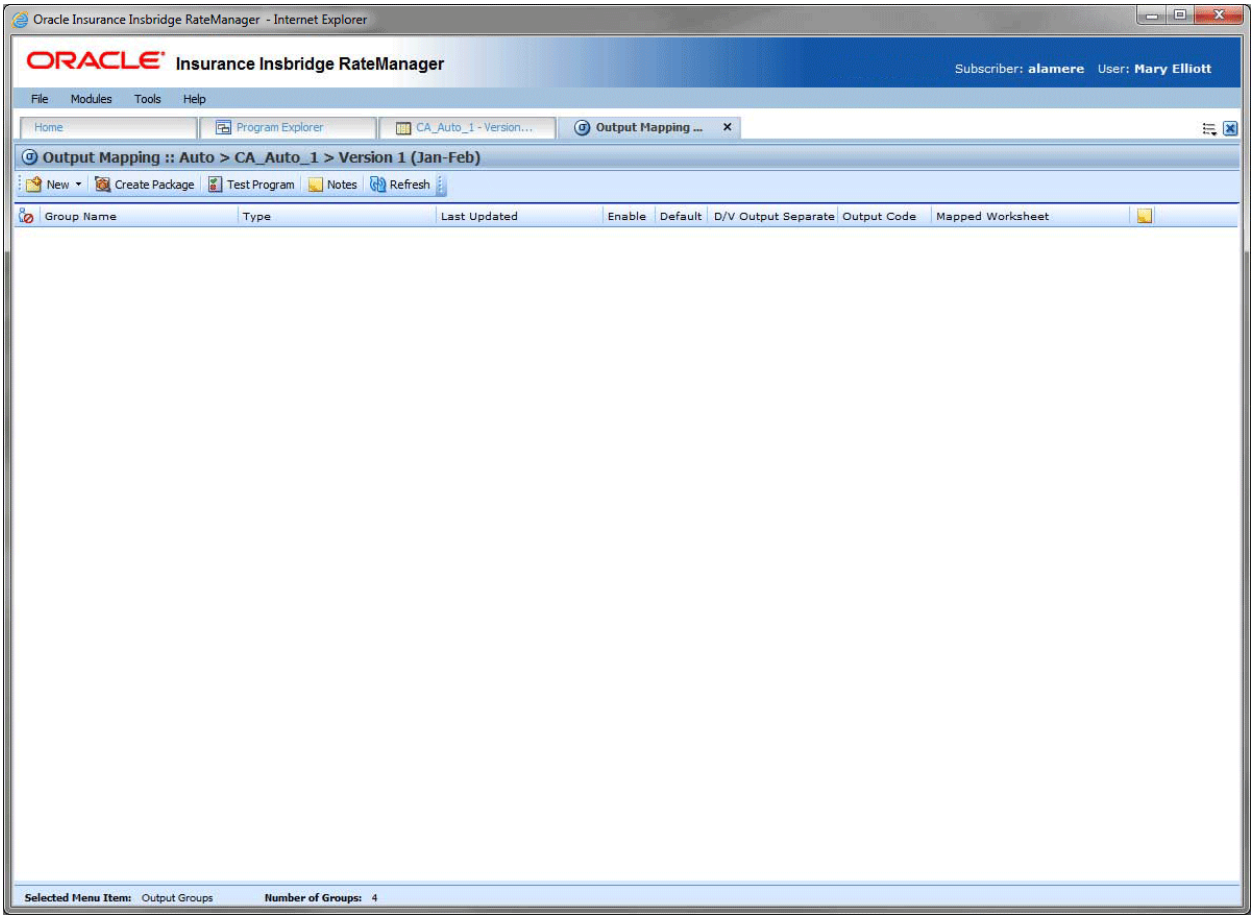
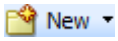
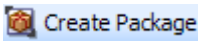
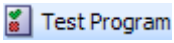
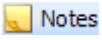
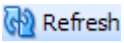


Figure 477 Output Mapping Tab

OUTPUT MAPPING FUNCTIONALITY

The **Menu Bar** located at the top of th**e** tab contains the functions you can perform.

	**New:** Select to create a new output group or a new worksheet.
	**Create Package:** Packages the program for rating and testing. See Packaging for more information.
	**Test Program:** Opens Test Case Editor where you can test and debug a program within RateManager. See Test Case Editor for more information.
	**Notes:** Enter, edit, and view notes to the output mapping.
	**Refresh:** Refreshes the screen with the current information.

OUTPUT MAPPING SCREEN DETAILS

Shows a listing of the output groups and worksheets created.

In Use Icon: An icon in this column indicates that this output mapping is being edited by another user.

Group Name: Name of the output group.

Type: The type of entry. Either an Output Mapping or a Worksheet.

Last Updated: Time stamp of when the output group was last saved or the options were changed.

Enable: A check indicates that the output group will be available to use when rating.

Default: A check indicates that the output group will be the default group used when rating, unless overridden in the field XML.

D/V Output Separate (Auto product only): A check indicates that output for the drivers and vehicles will be listed separately in the result XML. For more information, see the driver assignment topic Separating Driver and Vehicle Output (Auto product Only).

Output Code: An automatically generated code that cannot be changed but can be used to override the default output group. For more information, see the SoftRater section, SoftRater Switches.

Mapped Worksheet: Displays the name of the worksheet mapped to the output group.

Notes: A notes icon in this field designates that a Note has been added.

Right Click Menu for Output Mapping

Output Mapping Groups have a right click menu with some features that are not found on the menu bar.

- **Edit:** Opens the Edit Output Mapping tab where you can change the output mapping.
- **Copy:** Copies the selected output group.
- **Delete:** Deletes the selected output group.
- **Options:** Changes information about the selected output group. This includes the name, default option, enabled option and, if an auto program, the driver vehicle separate option.
- **Notes:** Opens up the notes screen where you can add, edit, view, and delete notes for the output group.
- **Map Custom Output XML:** Opens a window where you can create a mapping that will transform the output XML. See Input and Output Mapping for more information.
- **Program Output XML:** Opens a window that displays the Program Output XML.

CREATING A NEW GROUP

RateManager supports an unlimited number of output groups for any program version. Every program must have at least one output group. Each output group contains a specific set of elements that will be output when that group is used. The Output Mapping tab is where you can create, edit, copy, and delete an output group.

Naming standards can be applied to system elements. Naming standards allow for a uniform naming structure to be applied to system elements and programs. Naming standards include prefixes or suffixes attached to the element and for case enforcement to be applied to any newly created element or any existing elements that are being edited. If a naming standard is in place, the pattern is displayed in the element name field. If the naming standard is required, an error message is presented if the entry does not meet the required pattern. This is an optional feature and may not be presented for all elements.

NOTE: *Programs must be packaged before you can edit the group.*

To Create a New Output Group

1. Navigate to the **Output Mapping** tab for the program version where you want to create an output group.
2. Select **New→Output Group** from the menu bar. This opens the **New Output Group** window.

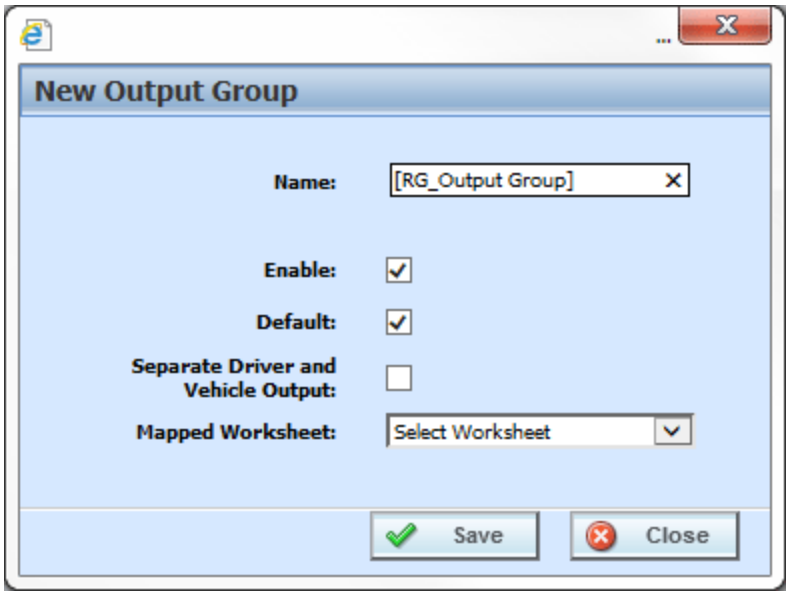


Figure 478 Entering a New Output Group

NOTE: *The Separate Driver and Vehicle Output option will only be available on Output Mappings in the Auto product.*

3. Enter a name for the new output group. If a naming standard is in place, a naming pattern is displayed in the name field. If the naming standard is mandatory, an error message is presented if your entry does not follow the pattern. Correct your entry and try again.

4. If this is the first output group, the **Enable** and **Default** options get checked automatically when saved. If this is an additional output group, the **Enable** option will get checked automatically when saved. To set the new group as the default, select the **Default** option.

5. **Separate Driver and Vehicle Output (Auto product only):** A check indicates that output for the drivers and vehicles will be listed separately in the result XML. For more information, see the driver assignment topic Separating Driver and Vehicle Output (Auto product Only).

6. Select the Mapped Worksheet you want associated with the output group. Worksheets must be mapped to be available and can be added at a later time. Worksheets are optional.

7. Click **Save** when you have finished to save the new output group and update the output group listing.

Default Output Mapping

The default setting signifies that this output mapping group will be used if no output mapping group is specified for the program version.

EDITING GROUP OPTIONS

Editing an output group is where you can change the name of an output group, as well as, whether it is enabled and/or the default. It is also where you can select a worksheet for the group. For an auto product output group, you also can set the separate driver vehicle option.

To Edit Group Options

1. Navigate to the **Output Mapping** tab for the program version that contains the output group you want to edit.
2. Select the group whose options you want to edit and then right click. Select **Options** from the menu. This opens the Edit Output Group window.

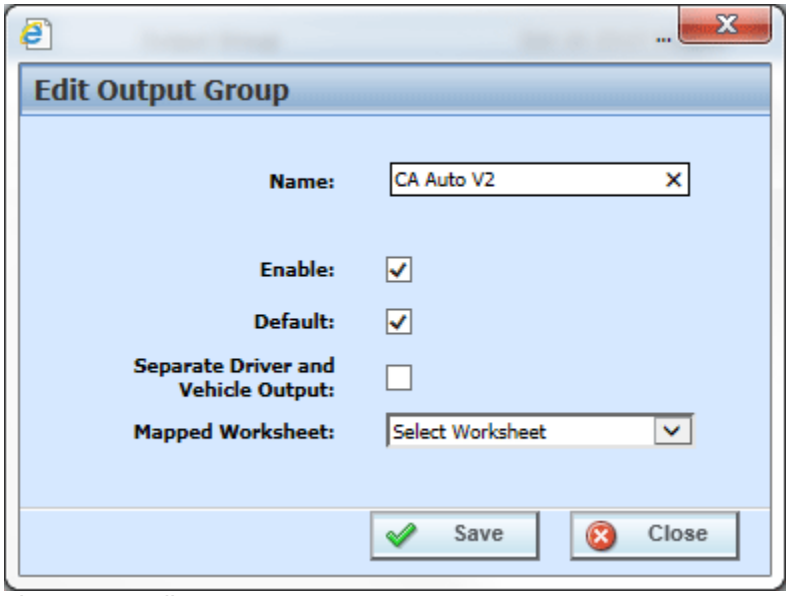


Figure 479 Edit Output Group

3. Make any updates and click Save.

Next Step

The next step is to create an output mapping. An output mapping defines the outputs from the program. Double click the output mapping group where you want to add a mapping. This opens the Edit Output Group Mapping tab.

EDIT OUTPUT GROUP MAPPING TAB

You can edit an output mapping when you want to change the output results or to change result IDs. The first time you enter the edit tab, it will be blank. You need to add variables based on the categories used.

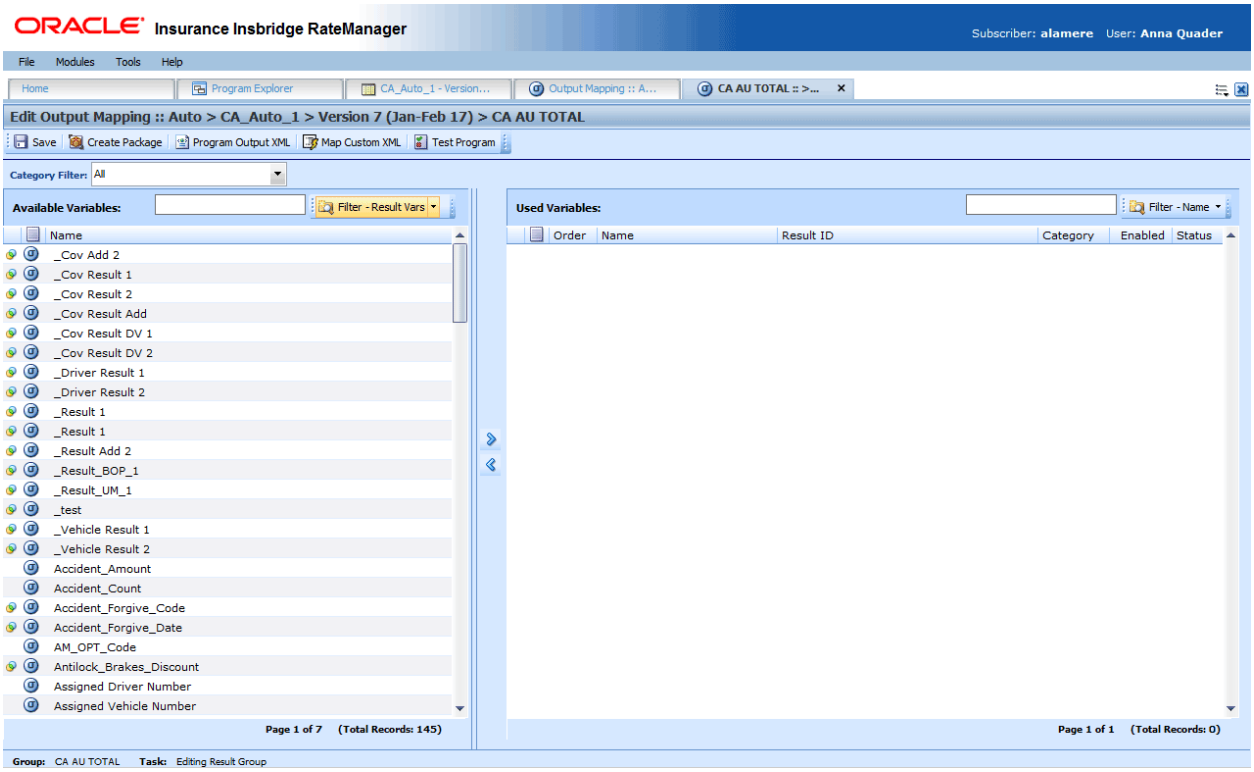


Figure 480 Editing an Output Mapping

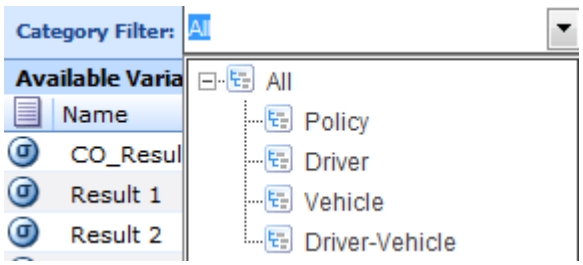
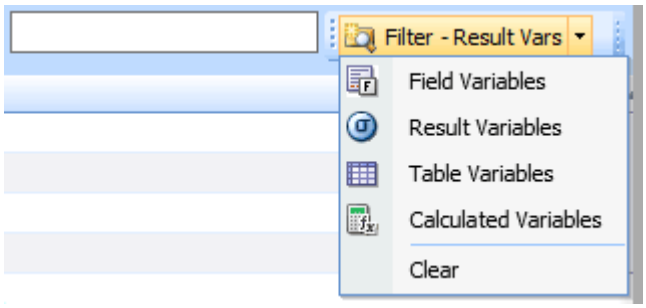
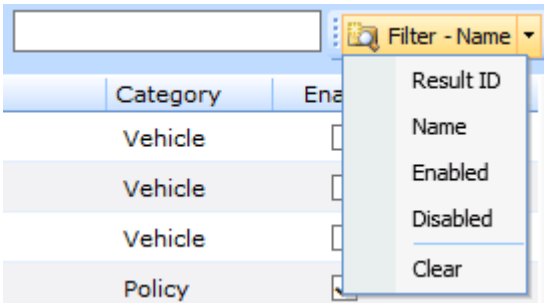
EDIT OUTPUT GROUP MAPPING TAB FUNCTIONALITY

The **Menu Bar** is located at the top of the **tab**. The menu bar contains the functions you can perform on this tab as well as search and filter options. The top bar menu has the options for creating packages, testing, and working with custom XML.

Save	**Save:** Saves the output mapping.
Create Package	**Create Package:** Packages the program for rating and testing. See Creating a Global Versioning Package for more information.
Program Output XML	**Program Outputs XML:** Opens a window that displays the Program Outputs XML.
Map Custom XML	**Map Custom XML:** Opens a window where you can create a mapping that transforms the output XML. See Input and Output Mapping for more information.
Test Program	**Test Program:** Opens the Test Program where you can test and debug a program within RateManager. See Test Program for more information.

Search and Filter

When there is a long listing of variables, it can be difficult to see if you have already used a variable or if the variable is available for use. You can define your search by availability and variable type.

	**Category Filter:** Filters the category used by the Available Variables and the Used Variables. Selecting another category causes the Available Variables to be updated with the variable type selected for the category selected and the Used Variables to display the results for the category selected. All is the default category. This selection displays all variables for the selected variable filter.
	**Available Variable Filter:** Filters the variables listed in the Available Variables column in alphabetical order. The filter options are for variable types. Clear clears the filter selection. Selecting another category causes the Available Variables to be updated with the category selected. Result is the default filter. The preceding box allows for a search against the variables listed.
	**Used Variable Filter:** Filters the variables that have been selected and are listed in the Used Variables columns. The filter options are for descriptive features. Clear clears the filter selection. Name is the default filter. The preceding box allows for a search against the variables selected.

	Remove variable
	Add variable

Navigation buttons are located in the middle of the tab. Use these buttons to move variables to the Used Variables area or delete variables from the Used Variables.

EDIT OUTPUT GROUP MAPPING TAB SCREEN DETAILS

There are two areas on the Output Mapping Tab.

- **Available Variables:** The list of the available fields, variables and results that can be used in the Used Variables area.

- **Used Variables:** A listing of fields, variables and results that have been selected. The listing shows the results in the order they will be outputted.

Edit Output Group Mapping Tab – Available Variables

Select the category you want and then select the available variable filter. The Available Variables area is populated with the available elements for the selection. To add an item to the Used Variables mappings, find the element you want to add in the Available Variables and double-click it. The element is added to the Used Variable area.

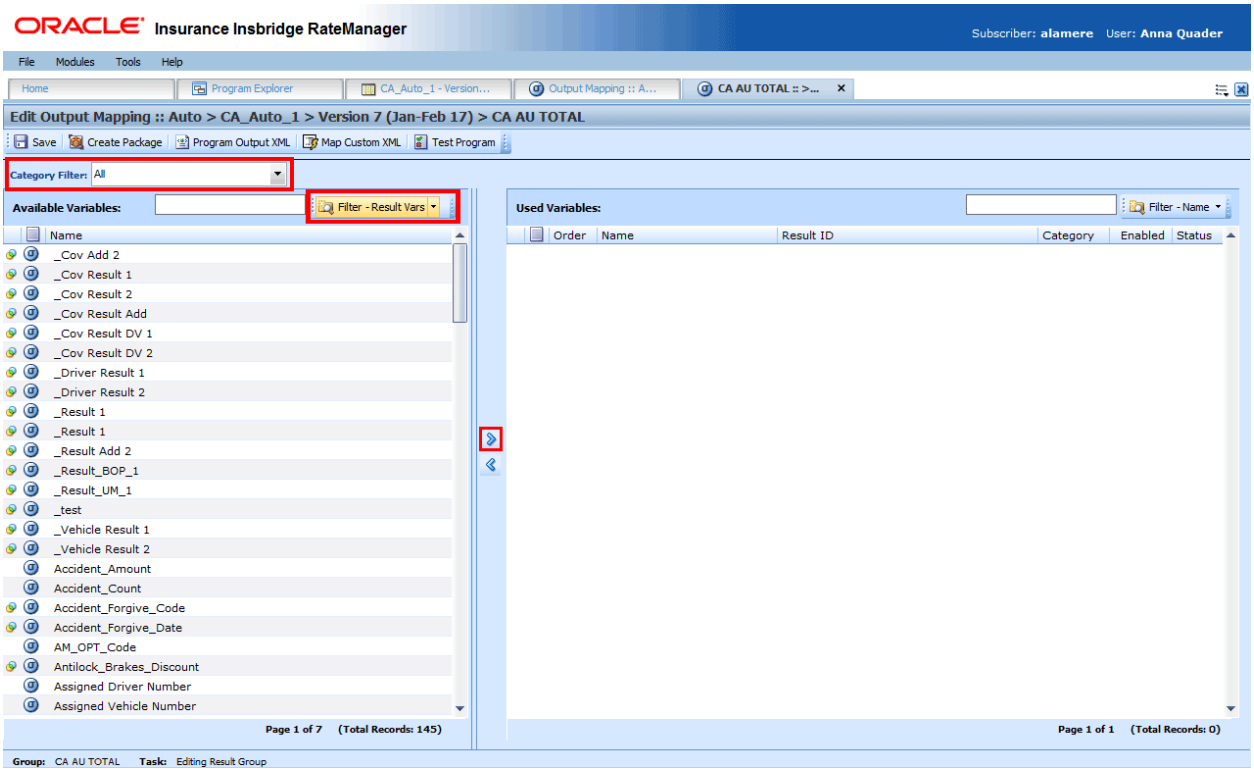


Figure 481 Available Variables for Output Mapping

Edit Output Group Mapping Tab – Used Variables

The Used Variables area contains a listing of fields, variables and results that have been selected for output. The listing shows the results in the order they will be outputted. The results can be moved by selecting, dragging and dropping.

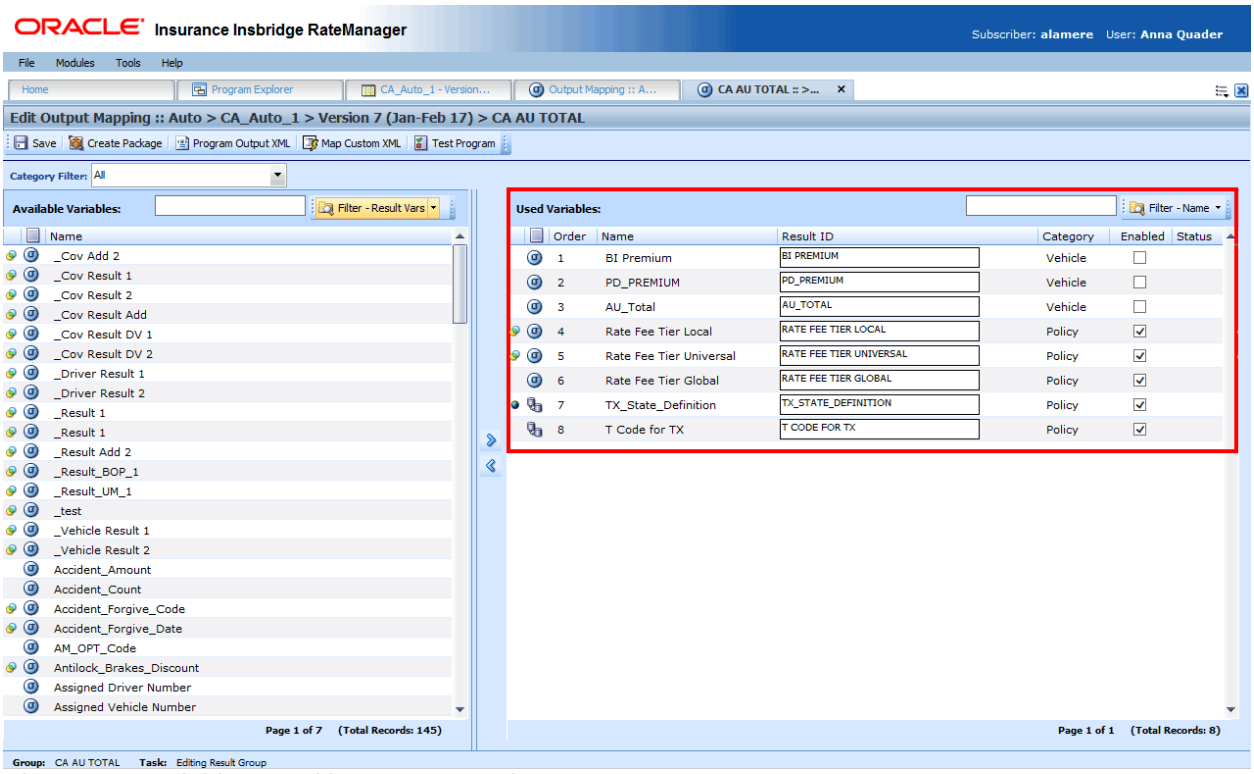


Figure 482 Variables Used in Output Mapping

Screen Details

Global Icon: Indicates if a global variable is being used.

Element Icon: Indicates the type of variable listed; Field, Result, Calculated Variable, or Table Variable.

Order: The order in which the result is displayed.

Name: The name of the variable.

Result ID: The associated ID that is used for XML mapping. Result IDs will automatically fill with the variable description in all capital letters. You can enter a different **Result ID** that you want in the associated field if necessary. The ID can be up to 256 characters.

Category: The working category of the variable or field.

Enabled: A check indicates that the result will be mapped. All results are active when brought into the selected column. A green checkmark will be in the **Enabled** column. If you want to leave a result in the listing, but do not want it executed, then remove the check next to the result.

Status: The current status of the variable.

Editing an Output Mapping

On the **Edit Output Mapping** tab you can choose which fields, variables, and result variables will be output in the rate result.

1. Navigate to the **Output Mapping tab** for the program version you want to edit.

2. Select the output group you want to edit and double-click it or right click and select **Edit**. This opens the **Edit Output Mapping** tab for the output group.

3. Select the category you want from the **Category Filter**.

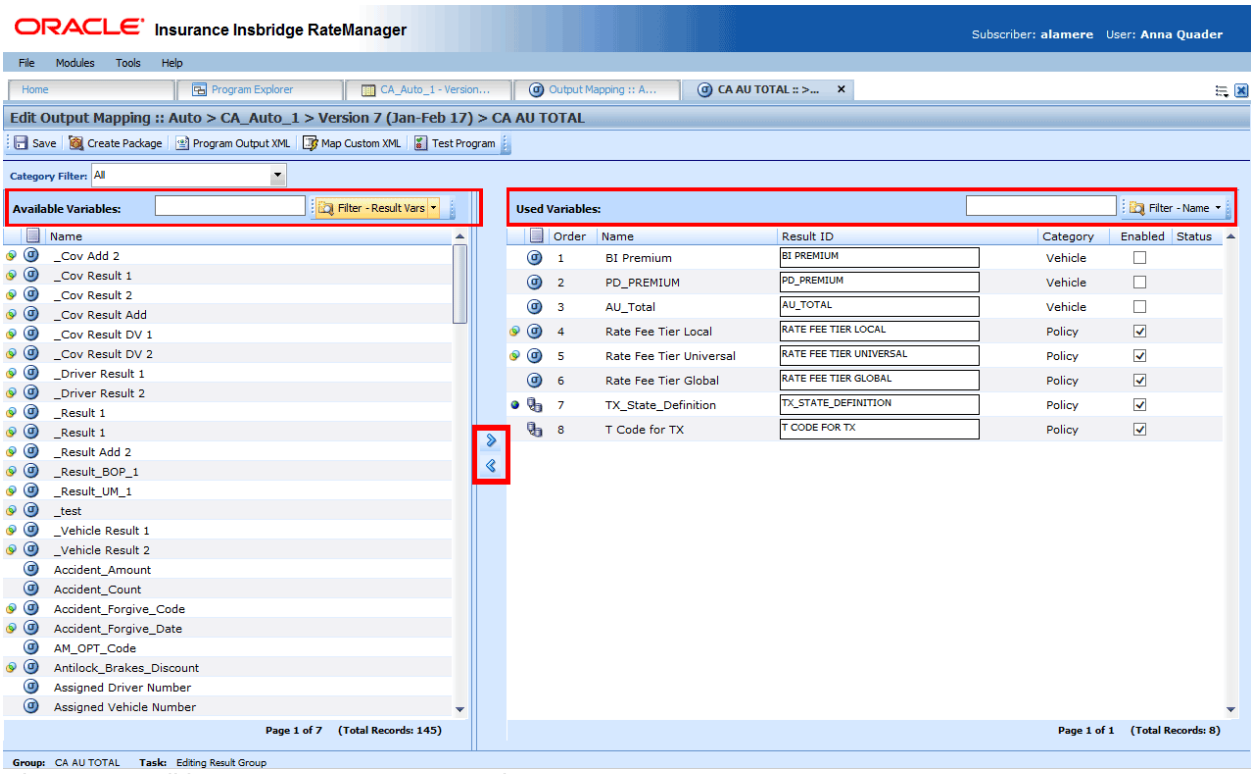


Figure 483 Editing an Output Group Mapping

4. Select the variable type you want displayed from the **Variable Filter**. The Available Variables listing will be populated with your selections.

5. Double-click the field, variable, or result you want to add. It will be added to the Used Variable listing. You can add as many fields, variables and results as you need. The first item you add to the selected column will be the first one listed, the second one added will be the second one listed and so on. You can change the order after you are done.

6. **Result ID**s automatically fill with the variable description in all capital letters. You can enter a different **Result ID** that you want in the associated field if necessary. **Result IDs** do not affect the rating process; they only exist to ease integration efforts with a company's existing back end systems. The ID can be up to 256 characters.

7. Make any edits that you may need. You can delete the elements you do not need and move the elements to the position you need as well. Make as many changes as you need

8. Click **Save** to save your work.

NOTE: *If you receive a message saying that a specific result cannot be added because the wrong category is selected, then this means that the working category of the result you are trying to add*

*differs from the filtered working category of the result mapping. Click **Yes** to automatically switch to the appropriate category and add the item to the result mapping.*

Multiple Selection Add or Remove

Multiple variables can be added to the mapping or removed from the mapping by holding the Shift key or Ctrl key when making your selections. The function is the same for the Available Variables side or the Used Variables side.

To Select a Range of Variables

Selecting a range of variables can be a fast way to create a mapping but may require moving variables to get the exact mapping results you want.

The first click you make while holding the Shift key marks the first selection. Keep holding the Shift key and make the second click. This marks the last selection. All selections in between to two are highlighted and will be added or removed.

If the variables are being added, the first variable selected is listed first, the last variable selected is next. The variables between the first and last are listed in the reverse order as the Available Variables listing.

If you are removing variables, the order does not matter and the variables are returned to the Available Variables listing in the same order.

To Select Variables in Sequential Order

Selecting the order of how you want your variables to be displayed on the output can be a faster method of creating a mapping as well. The variables will show as selected but do not indicate what order they have been selected in.

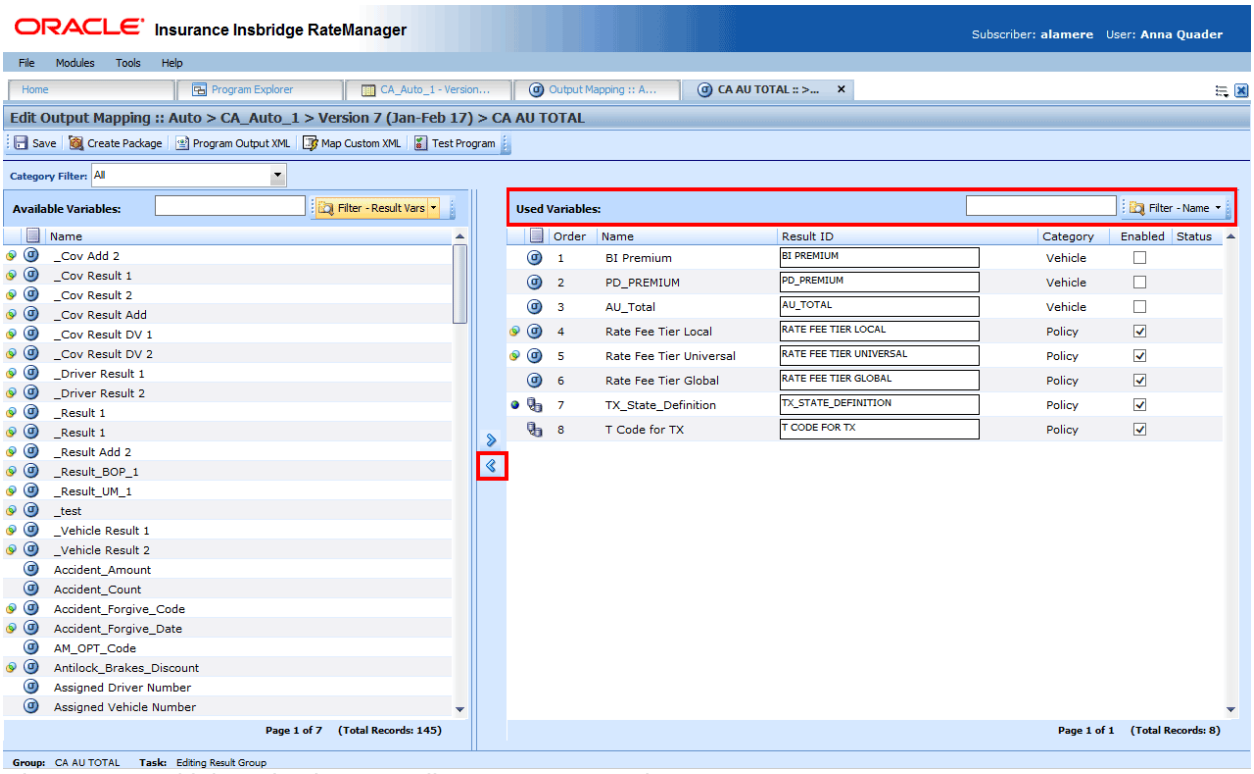


Figure 484 Multiple Selections to Edit an Output Mapping

If you are creating a mapping, hold the Ctrl key down and click the first variable you want in your mapping. Click the next variable you want to run and continue until you have the mapping order you want. Click the Move button to move your selections over to the Used Variables.

If you are removing variables, hold the Ctrl key down and make your selections. Click the Remove button to move your variables over to the Available Variables. The order does not matter and the variables are returned to the Available Variables listing in the same order as they were removed.

A variable can be removed by holding the Ctrl key and clicking the variable a second time. The variable is no longer highlighted and no longer included in the output mapping.

Changing Output Order

To change the result order, highlight the variable you want to move. Hold down the mouse button to drag and drop the variable to the position you want. You will know the result is ready to move when you see a shadow box containing the result information displayed.

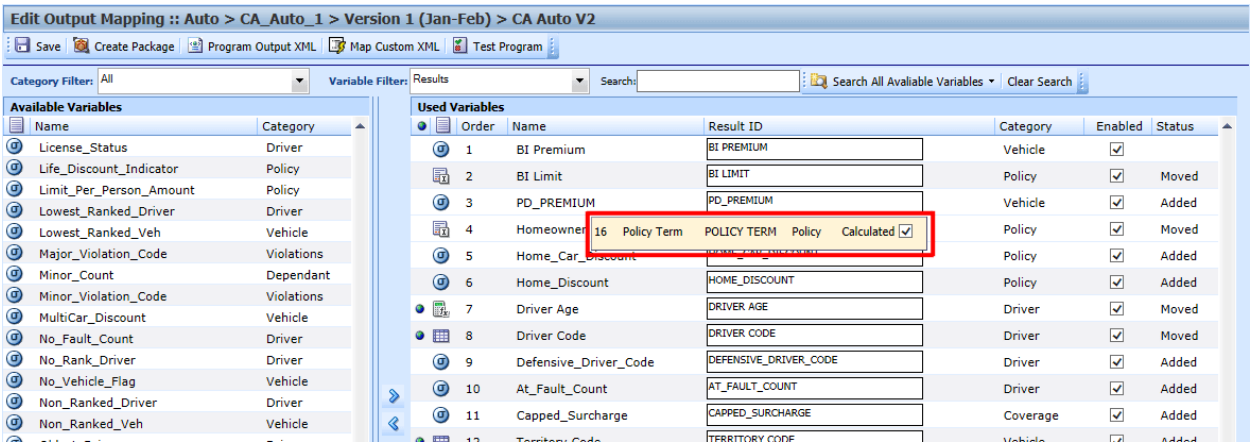


Figure 485 Moving Output Mapping

Output Code Location

Output codes are required for custom XML mappings. This overrides the default result mapping group. By default, SoftRater uses the group that is set up as the default in RateManager. To use a different result group, you will need to add the attribute result_def="Result_Code" where Result_Code is the output code shown on the Output mapping tab.

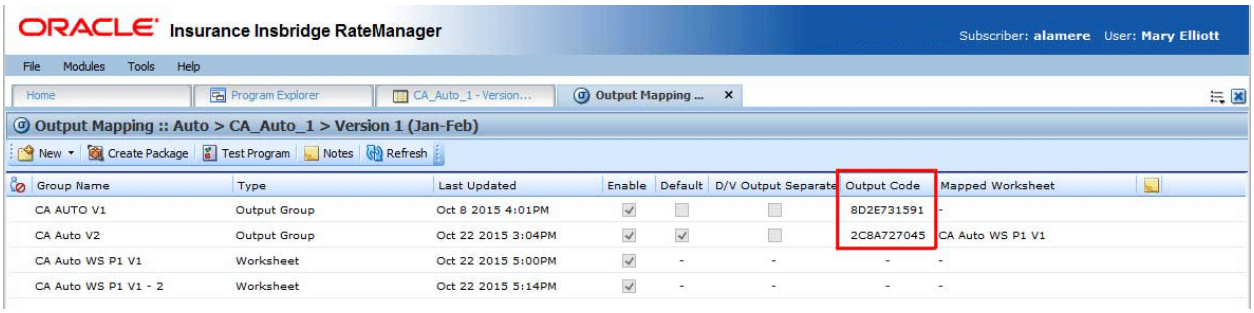


Figure 486 Output Codes

If you are not using custom XML mappings, you will not need to enter the output code anywhere.

DELETING A GROUP

Output groups can be deleted at any time. It is possible to delete all output groups. However, in order to package a program, you must have at least one output group.

To Delete a Output Group

1. Navigate to the Output Mapping tab for the program and version that contains the output group you want to delete.

2. Select the group and then right click. Select **Delete**. A warning message will be displayed.

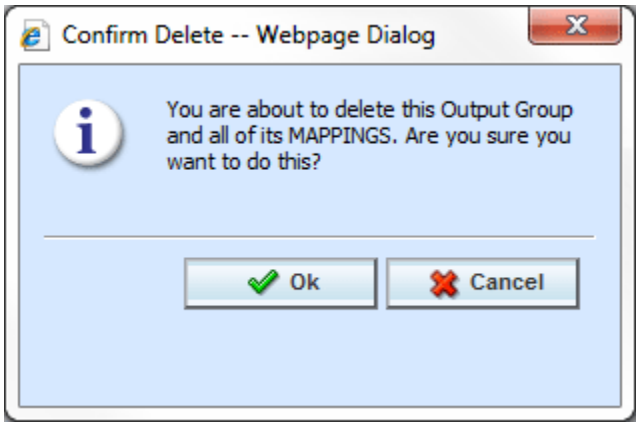


Figure 487 Deleting an Output Group Mapping

3. Select yes to remove the output group. Select no to return to the Output Mapping tab without deleting an output group.

RATING WORKSHEET

Insurers may elect or be required to provide a Rating Worksheet or Premium Calculation Worksheet with each issued policy that explains the information used to establish rates and premiums. The needs and requirements for the format of these types of documents can vary greatly. States have very specific instructions and requirements on the details required in a Rating Worksheet or Premium Calculation Worksheet. Oracle Insurance Insbridge Enterprise Rating has made the data available in a format that allows you to create documents that meet requirements.

RateManager has the ability to create a Rating Worksheet. A Rating Worksheet in RateManager is an XML document that conveys the basic information used to establish individual rates and premiums. The output is based on the current value of the terms at that point in the execution of the step. For example, if result A is being used in step 3 of algorithm 1 and then again in step 5 of algorithm 10, the value of result A will be shown twice in the worksheet – once for algorithm 1 and once for algorithm 10. You select the steps where you want to view the current values. The output reflects the current value at the point of execution of the algorithm. The XML document can be carried into most existing systems where it can be used and formatted in a report to fit your needs.

A worksheet is created in Output Mapping and can be carried through to Test Case Editor, Testing, Impact Analysis, and SoftRater Test Interface in IBFA and IBSS.

Worksheet Relationships

- Worksheets are program version specific. A worksheet created in version 1 is not available to version 2.
- There is no default worksheet. Worksheets must be mapped to an Output Group. If the Worksheet XML is completely blank, then no worksheet has been mapped to the Output Group.
- You must check the algorithm steps that you want returned in the Worksheet. You can select all steps by clicking the steps icon at the top of the field. If the worksheet XML only has header information, then no steps were selected.
- The algorithms listed on the worksheet are updated when the sequence is changed.
    - If an algorithm is added to the sequence, it will be available to be used in the worksheet. The added algorithm will not be automatically checked.
    - If an algorithm is removed from the sequence, it is also removed from the worksheet listing.
    - If changes are made to the program that the worksheet is mapped to, then the worksheet may need to be re-mapped to incorporate the changes.
- In Testing and Impact Analysis, worksheet changes are applied to every package that uses that worksheet. Changes to a worksheet require that a new package be created. If you rate a program in Testing and then update a worksheet, you will need to create a new package. After you create a new package you can re-rate the program, and the updated worksheet will be used. Results from the previously run program will not be changed.

CREATING A WORKSHEET

Prior to creating a worksheet, make sure that sequencing is complete. Worksheet algorithms are the ones that are used on the Sequence screen. If there are no selections on the sequence screen, there are no options on the Edit Worksheet screen.

Worksheets can be created before or after the output group. If created before the output group or before the program is ready, you may want to enable the worksheet at a later date.

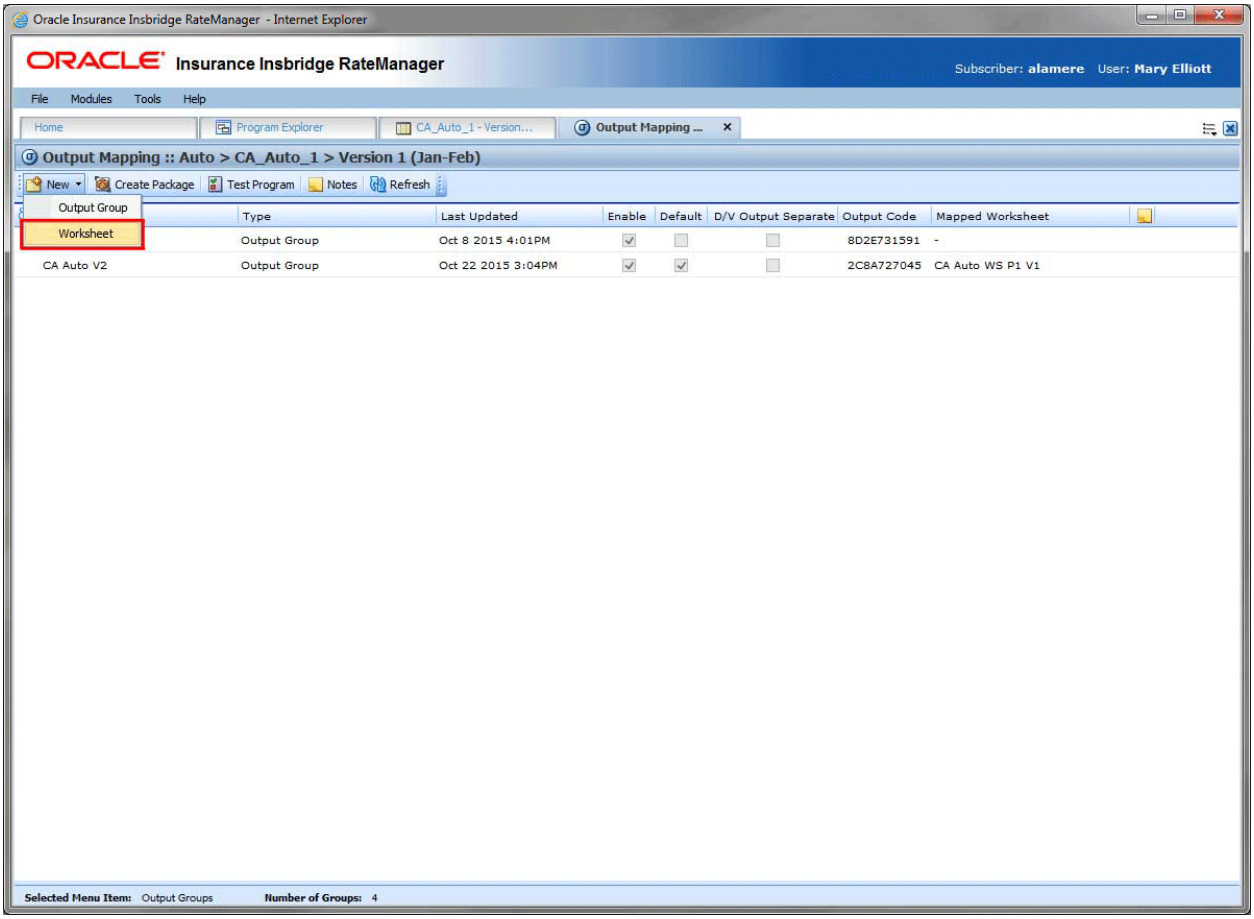


Figure 488 Creating a New Worksheet

Rating Worksheets are program version specific and created on the Output Mapping tab of a program.

1. Select the program version where you want to create a worksheet.
2. Open the Output Mapping tab.
3. Click New and select Worksheet. A separate screen is displayed.
4. Enter a name and select if the Worksheet should be enabled.

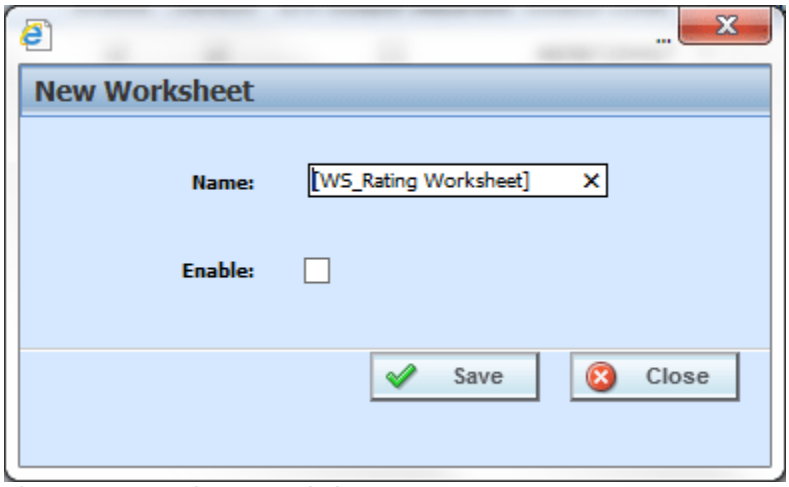


Figure 489 Naming a Worksheet

5. Click Save. The Output Mapping screen is presented and the new worksheet is listed.

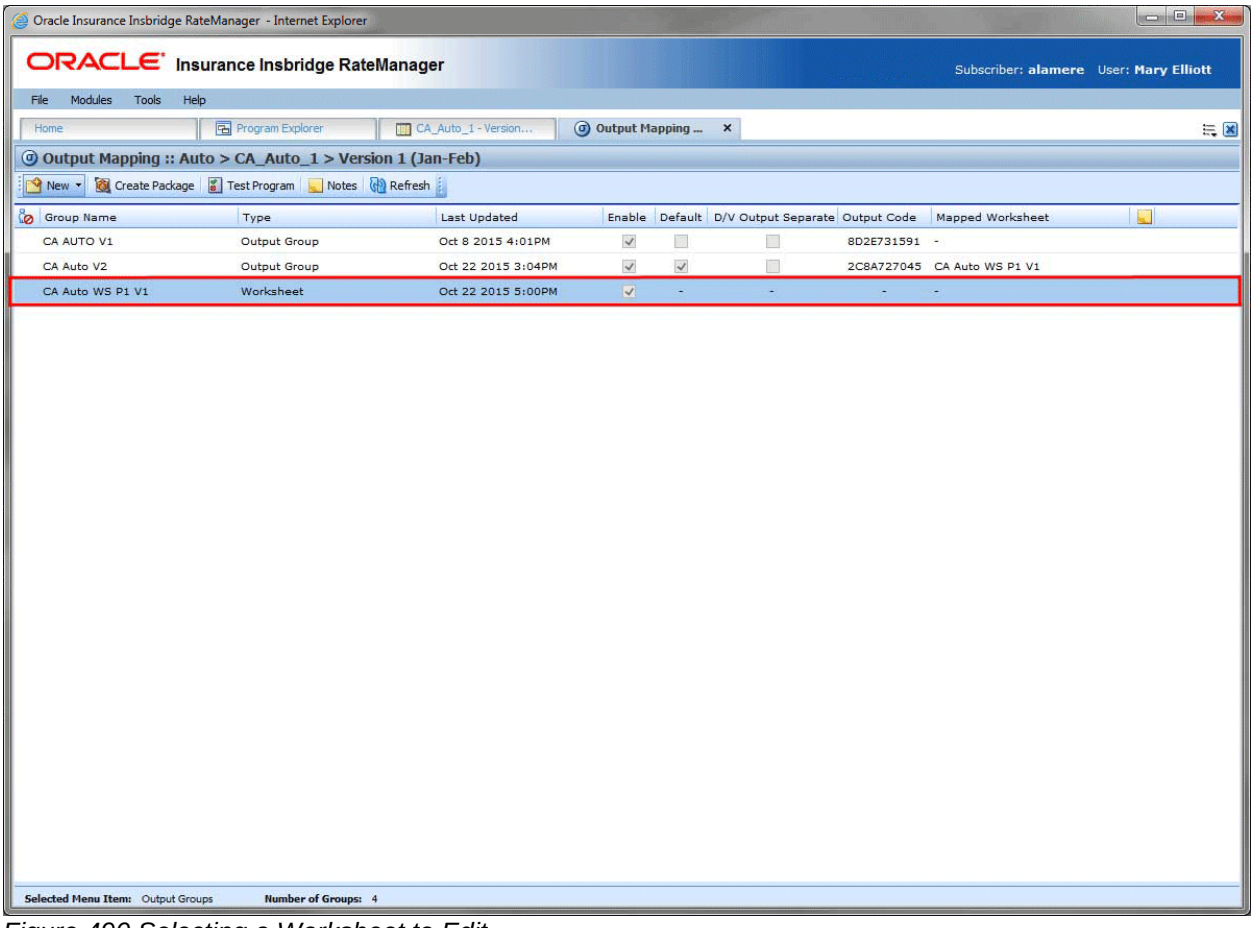


Figure 490 Selecting a Worksheet to Edit

Worksheet Right Click Menu

Every worksheet has a right click menu available.

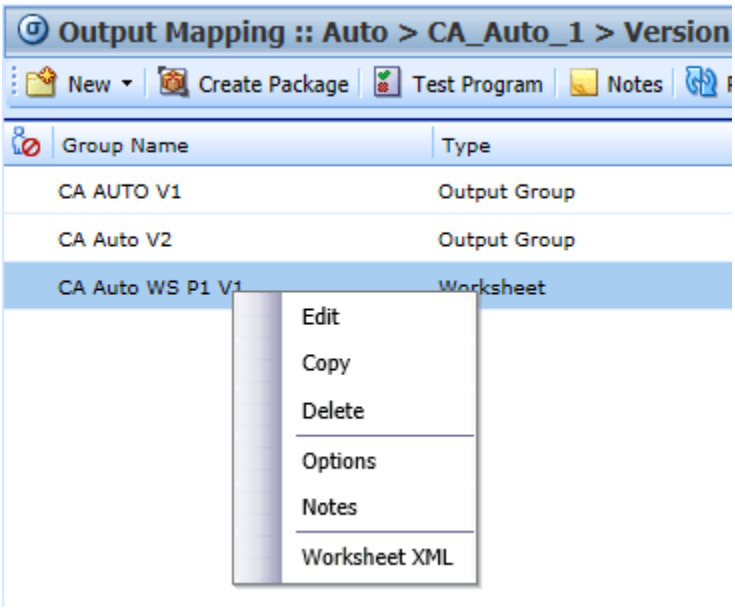


Figure 491 Right Click Menu for Worksheets

The right click menu options are:

- **Edit:** Opens the Edit Worksheet screen. Used to add algorithms to the worksheet.
- **Copy:** Creates a copy of the worksheet. All details remain the same. The name must be updated.
- **Delete:** Deletes the selected worksheet.
- **Options:** Open the options window. Use this option to enable or disable the worksheet and to edit the worksheet name.
- **Notes:** Allows for notes to be added to the worksheet. Notes stay local and are not carried out to the XML.
- **Worksheet XML:** Opens the Worksheet XML screen. This view reflects the current selections of the worksheet. The worksheet XML screen cannot be minimized. But the same design time output worksheet XML can be minimized through the edit worksheet window.

Editing a Worksheet

When a worksheet is first created, no algorithms are checked to be included. You must edit the worksheet. Editing a worksheet allows you to select the algorithm steps that you want to include. You can select as you like.

Editing is for the selection of algorithms only. If you want to enable the worksheet or update the name, use the Options option on the right click menu.

To Edit a Worksheet

1. Stay on the Output Mapping tab.

2. Double click to edit or right click and select Edit from the menu. The Worksheet screen is displayed.

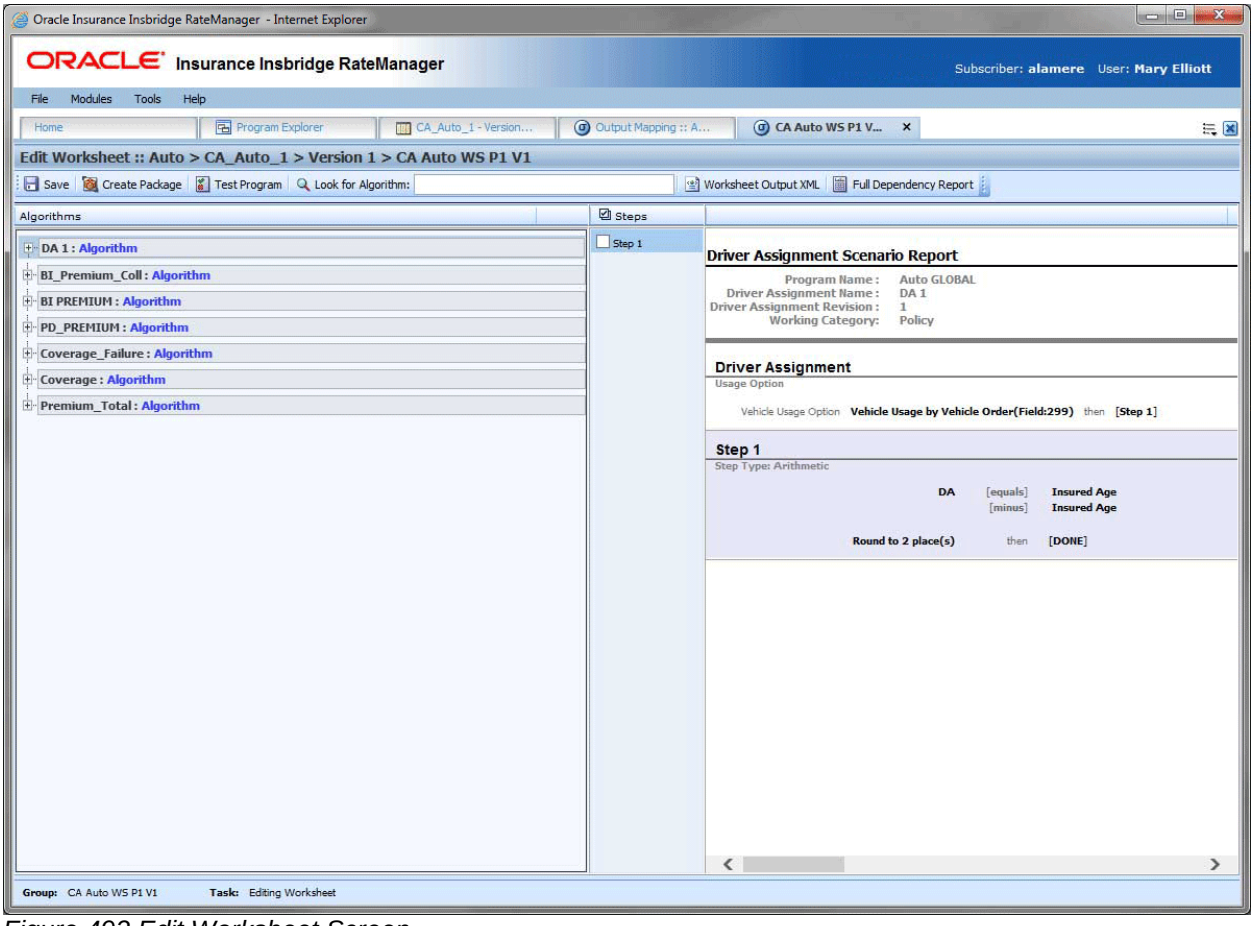
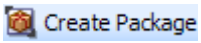
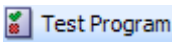
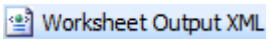
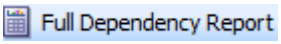
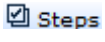


Figure 492 Edit Worksheet Screen

Worksheet Functions

	Save: Saves the worksheet. If you makes changes and exit without saving, you will be prompted to save.
	**Create Package:** Packages the program for rating and testing. See Packaging for more information.
	**Test Program:** Opens Test Case Editor where you can test and debug a program within RateManager. See Test Case Editor for more information.
	Look for Algorithm: Searches the algorithms for the matches to criteria entered.
	**Worksheet Output XML:** Displays the worksheet output XML.
	**Full Dependency Report:** Displays a Full Dependency Report.
	**Steps:** Located at the top of the steps field. Clicking the steps icon selects all the steps in the selected algorithm.

Worksheet Screen Details

There are three areas:

- **Algorithms:** Lists the algorithms selected on the Sequencing screen.
- **Steps:** The number of steps in the selected algorithm.
- **Reports:** Reports are displayed for algorithms, underwriting rules and drivers assignments. The Reports area displays a preview of the selected algorithm. By default, the first algorithm is displayed.

Algorithms

The far left area contains the algorithms selected on the Sequencing. Algorithms are listed in the same order. No algorithms are checked when you first enter. Algorithms are presented in the same way as a Dependency report.

You can expand the algorithm out to view the details by clicking the plus sign at the front of the algorithm.

The selected algorithm is highlighted in blue.

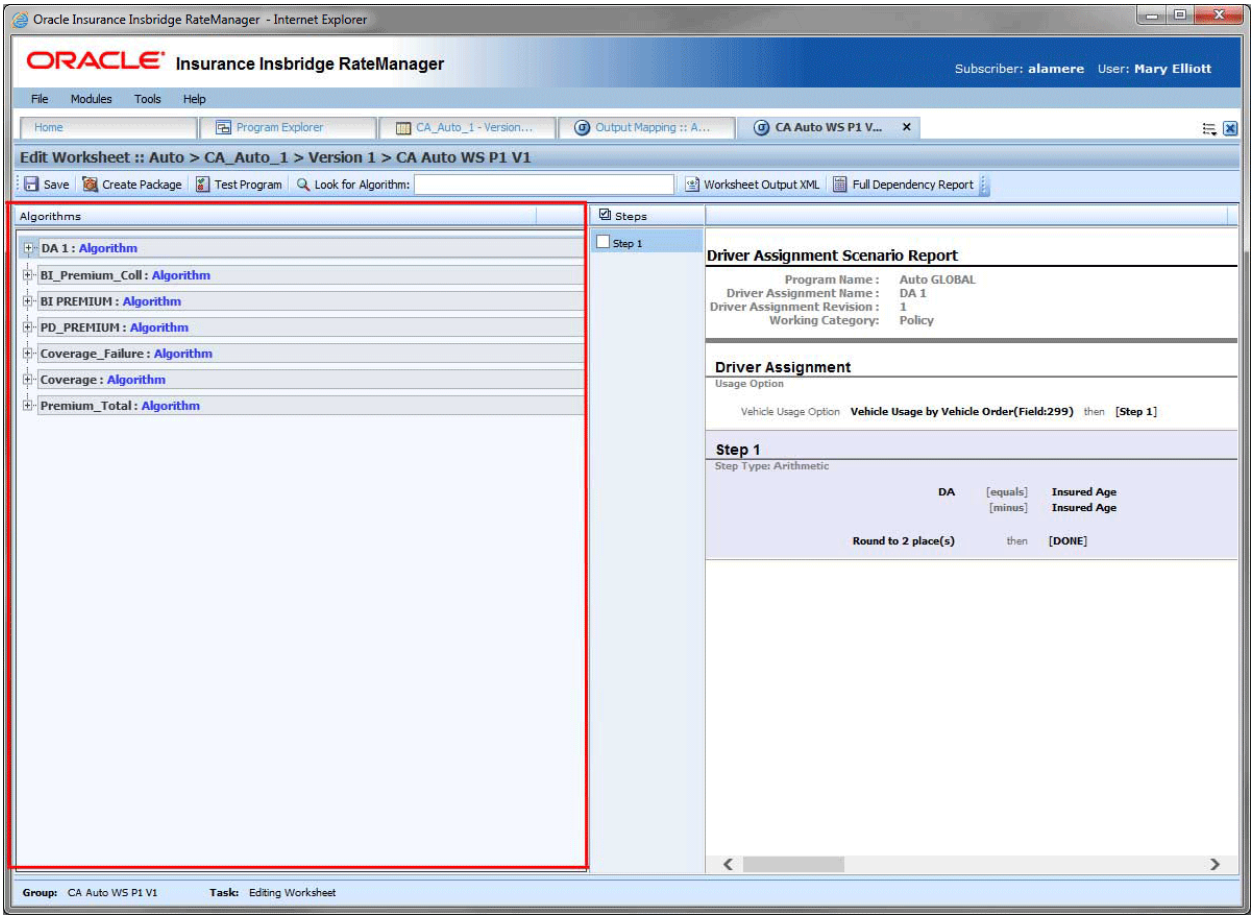


Figure 493 Edit Worksheet Algorithms

Steps

The steps column in the middle of the screen lists the number of steps found in the algorithm. To include or remove all the steps in the algorithms, click the Steps icon in the top of the column. This selects all steps. You also can select the individual steps you want to include.

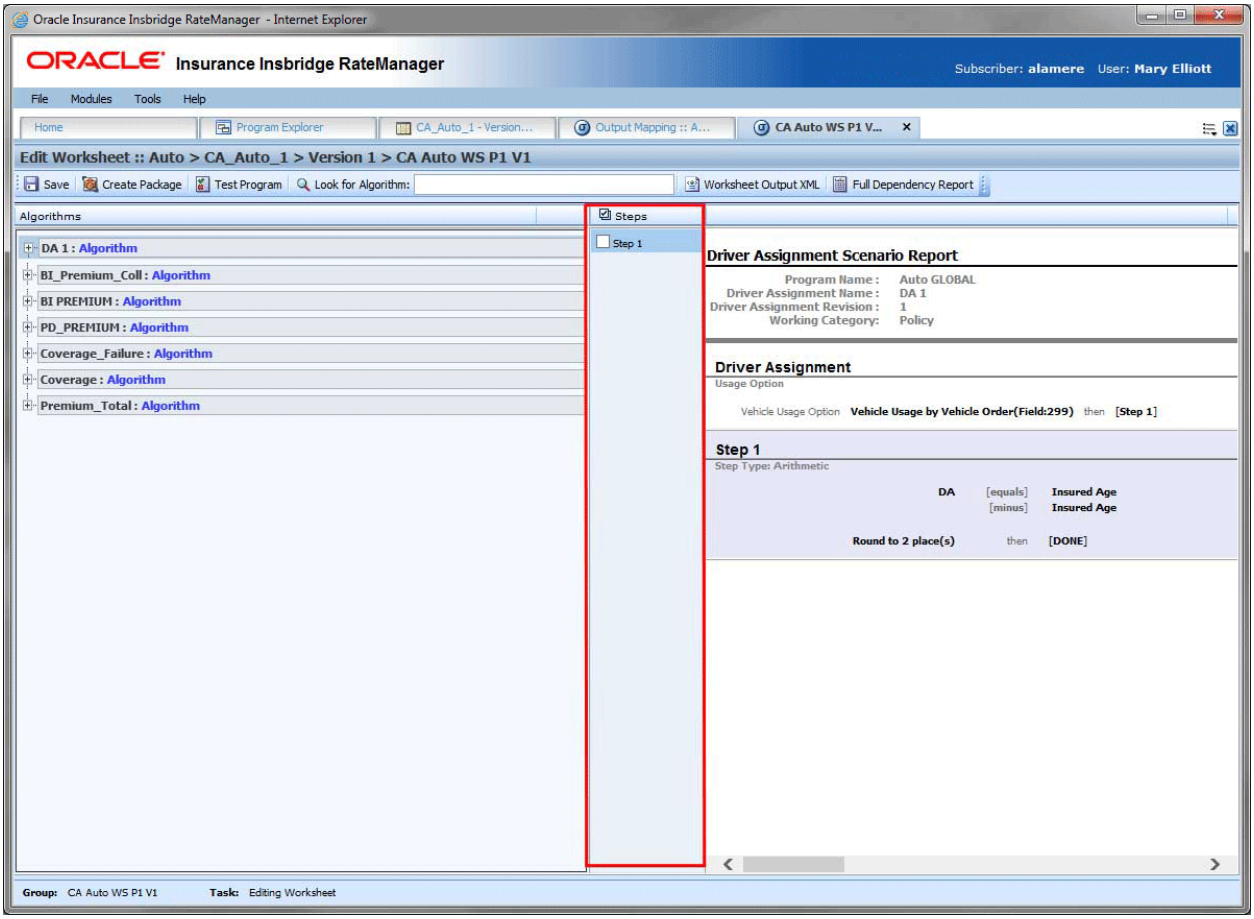


Figure 494 Edit Worksheet Steps

Reports

The right side of the screen contains the algorithm report for the selected algorithm, underwriting rule, or driver assignment. This is an information only screen that contains the details of the algorithm.

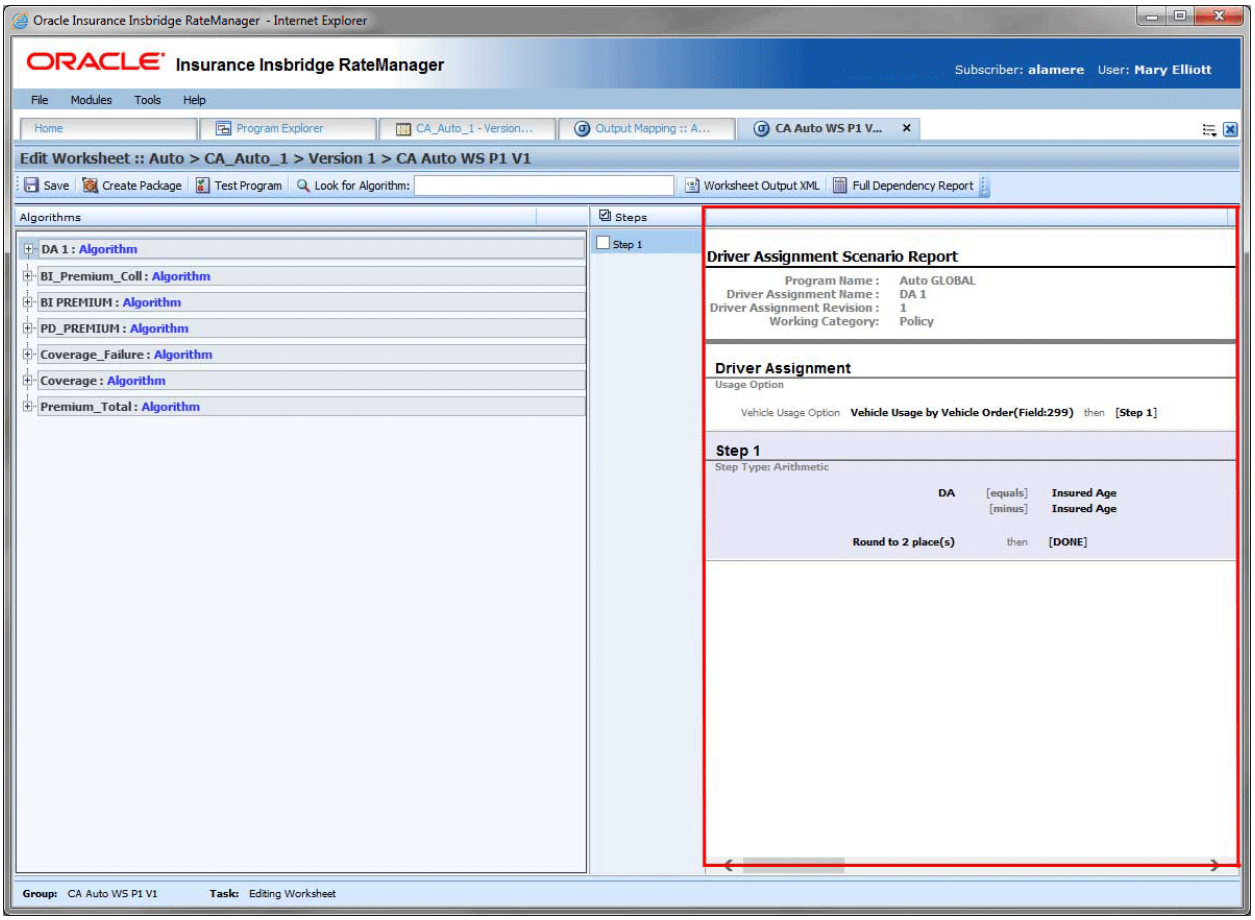


Figure 495 Edit Worksheet Report

Adding Selections to a Worksheet

1. Select the algorithm that you want to add to the worksheet. The algorithm is highlighted in blue. The steps are listed in the steps column and a report is presented in the reports area. Algorithms are not expanded when you first enter. Click the plus sign in front of the algorithm to expand the algorithm to view the details.

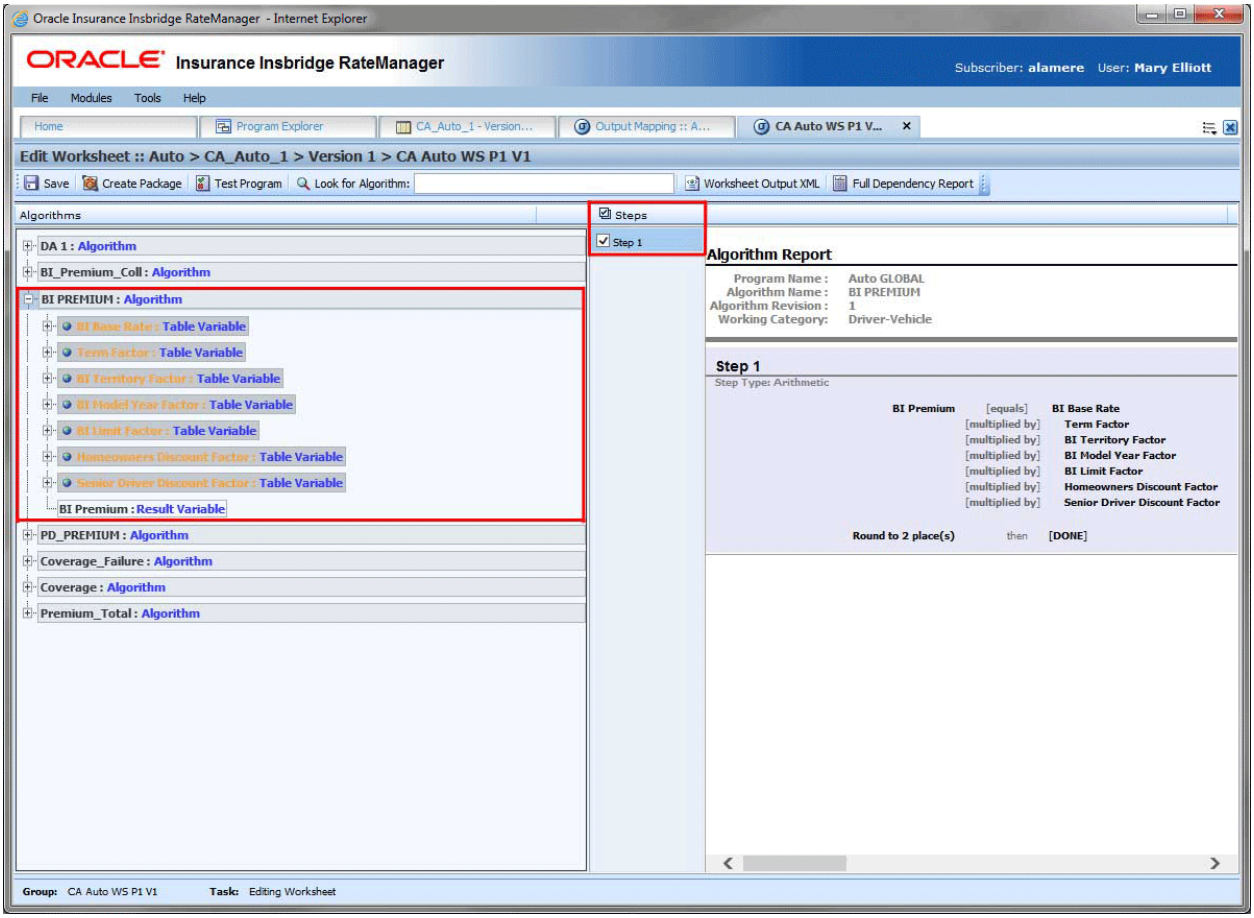


Figure 496 Selecting Algorithm Steps for Worksheet

2. Check the steps you want included.

3. Click Save.

4. If you are ready to test, create a package. The worksheet will not appear in the debug report on Testing unless a package is created after the worksheet is saved.

NOTE: *Saving does not validate that any selections have been made. It is possible to save a worksheet with no selections.*

After saving the worksheet, there are two different options for viewing:

- **Worksheet Output XML**
- **Full Dependency Report**

Worksheet Output XML

After you have saved your worksheet, you can preview the worksheet XML by clicking the Worksheet Output XML. The Output XML displays the selected algorithm steps in XML format. You can view or copy.

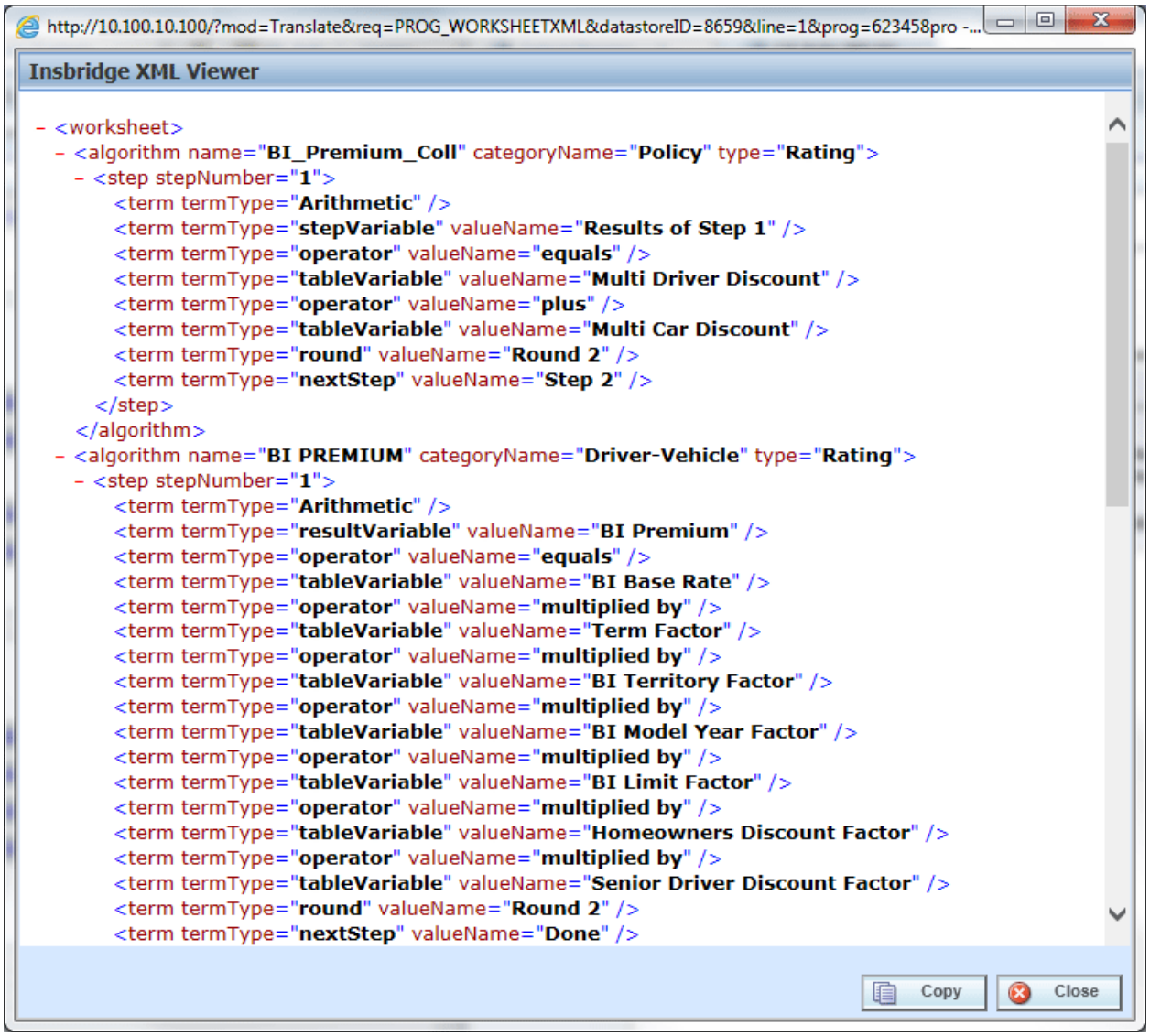


Figure 497 Worksheet XML Viewer

If the output is “No Worksheet information available,” check to make sure you have checked an algorithm and saved your entry.

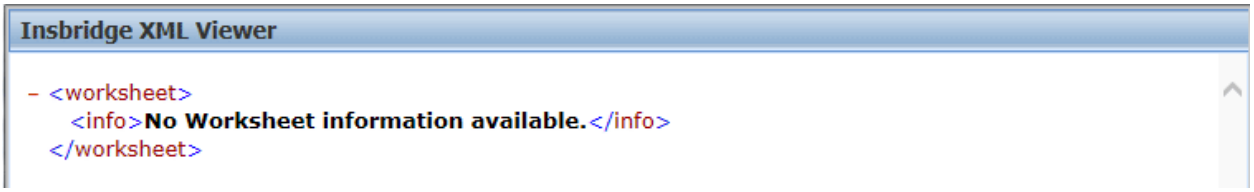


Figure 498 XML Worksheet Error

Full Dependency Report

A Full Dependency Report presents the relationships between variables in output. You can view the algorithms, underwriting rules, and drivers assignments used. Details regarding the variables used within these elements; calculated variables, tables, inputs, and results can be viewed by expanding the variable.

For more information, please see Reports.

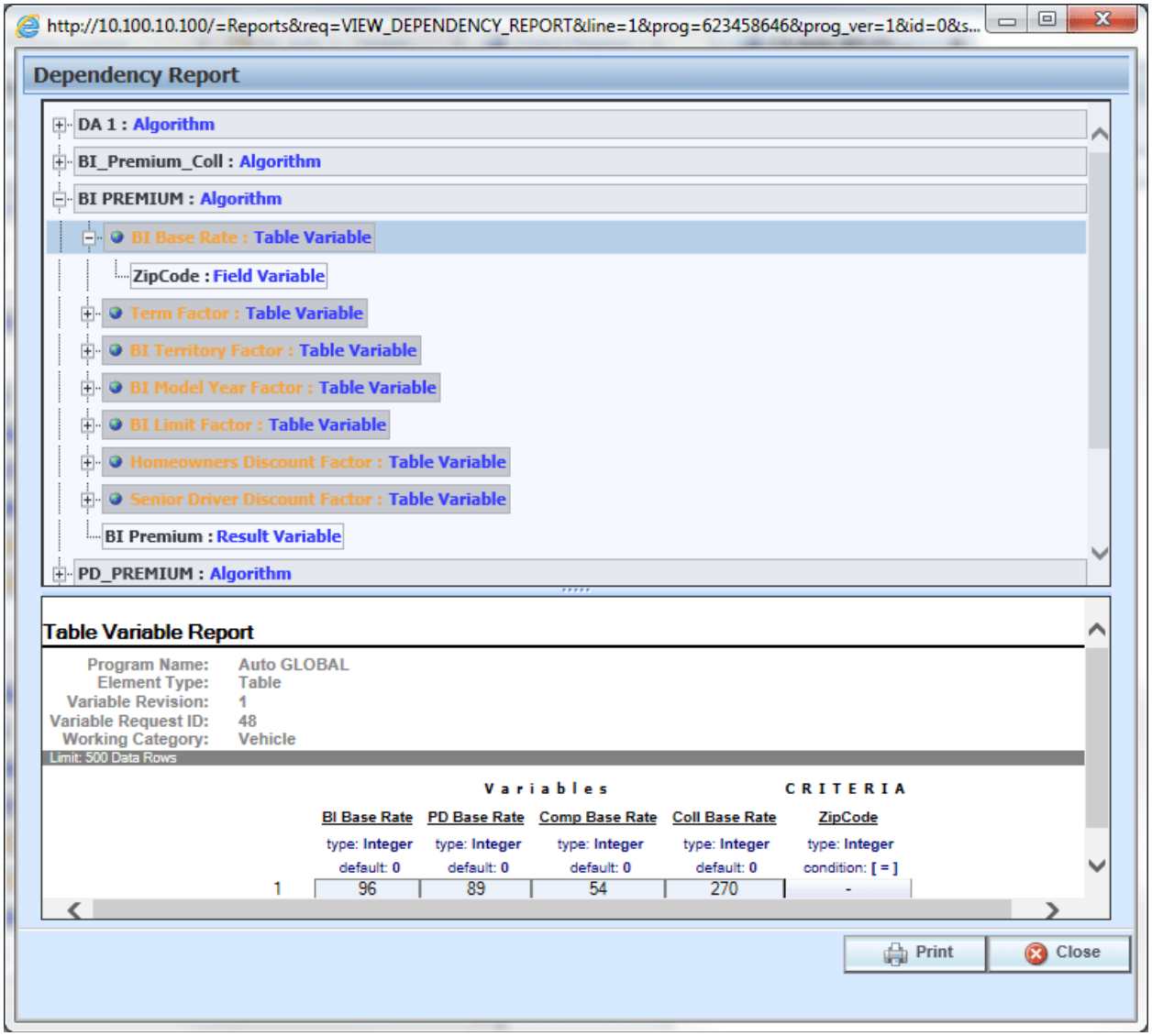


Figure 499 Worksheet Dependency Report

NOTE: *The Insbridge message service must be on for the reports to generate.*

Enabling/Disabling a Worksheet

Enabling a worksheet allows for the worksheet to be included in the output group. If you want to view worksheets in Test Case Editor, a system administrator must allow for it. If a worksheet is included in the output group, it can be exported out of the Insbridge system as an XML and used by other systems.

A worksheet can be enabled or disabled at any time. You may not need worksheet information until the program is finished. You may want to wait to enable a worksheet until after developing the program in order to reduce package size. You may want to disable a worksheet if the information is no longer needed.

If you enable or disable at a later time, you may need to create a package for the worksheet to be included or removed.

To Enable/Disable a Worksheet

1. Select the worksheet that you want to enable.

2. Right click and select Options. The Edit Worksheet screen is displayed.

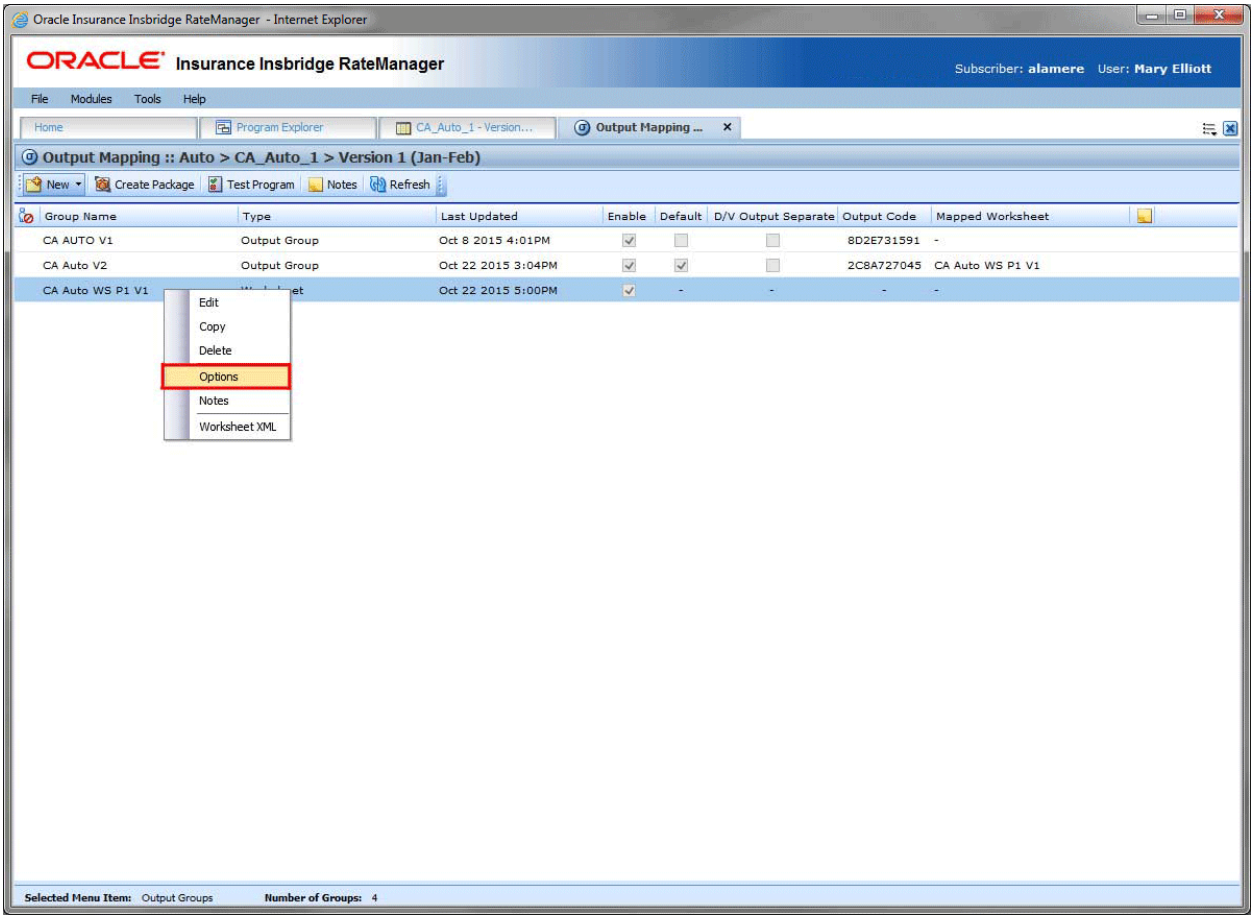


Figure 500 Worksheet Options

3. Check the Enable box. If the Enable box is checked, this enables the worksheet and allows for it to be selected by an output group. If the Enable box is unchecked, this removes the worksheet

from selection by an output group. It also stops a worksheet from being created after the program is packaged.

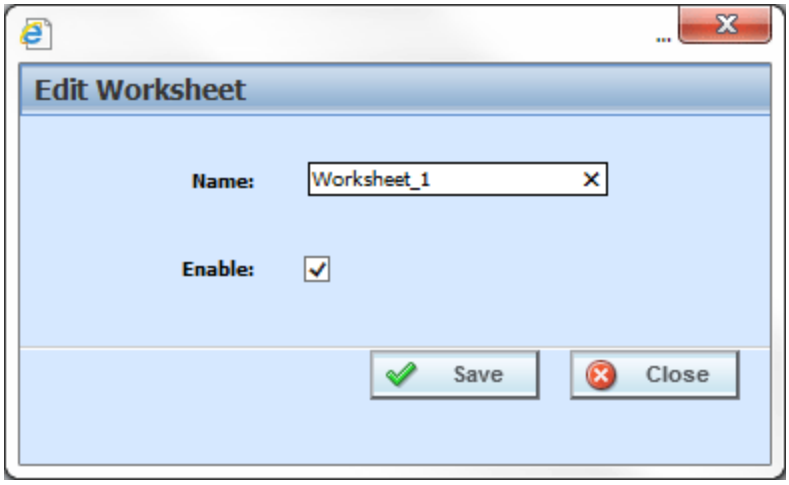


Figure 501 Enabling Worksheet

4. Click Save. You are returned to the Output Mapping tab. The worksheet now shows as enabled or disabled.

Editing a Worksheet Name Using Options

Any worksheet can be edited at any time using the Options option. The Options option is available only on the right click menu of the worksheet on the Output Mapping tab. Options allow you to edit the name of the worksheet and to enable or disable the worksheet.

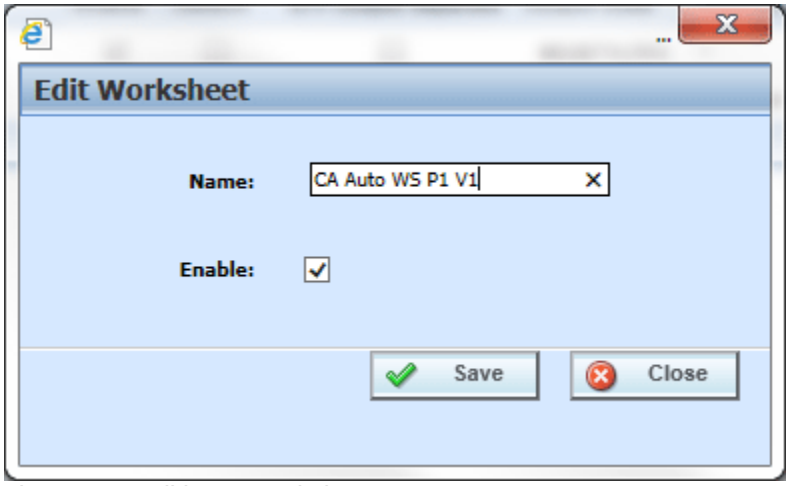


Figure 502 Editing a Worksheet

1. Select the worksheet you want to edit.
2. Right click and select Options. The Edit Worksheet screen is displayed.
3. Enter in a new name.

4. Check to enable or disable.
5. Click Save.

Copying a Worksheet

Any worksheet can be copied at any time. All selections are copied, including if the worksheet is enabled. Worksheet names must be unique per program version.

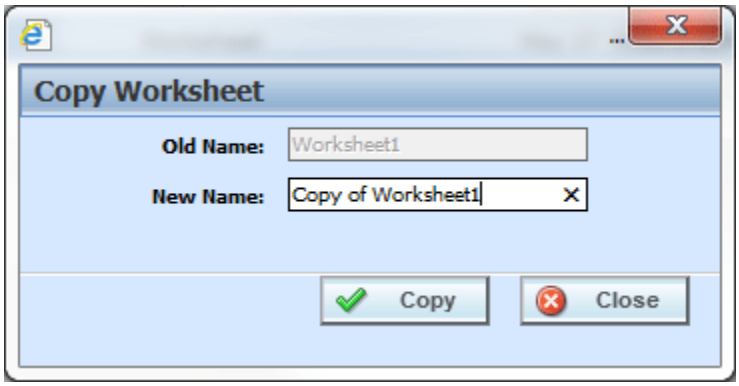


Figure 503 Copying a Worksheet

1. Select the worksheet you want to copy.
2. Right click and select Copy. The Copy Worksheet screen is displayed.
3. Enter in a new name.
4. Click Copy. The copied worksheet is displayed on the Output Mapping tab and available for editing.

Deleting a Worksheet

A worksheet can be deleted at any time. If a worksheet is mapped to an output group, the association is removed.

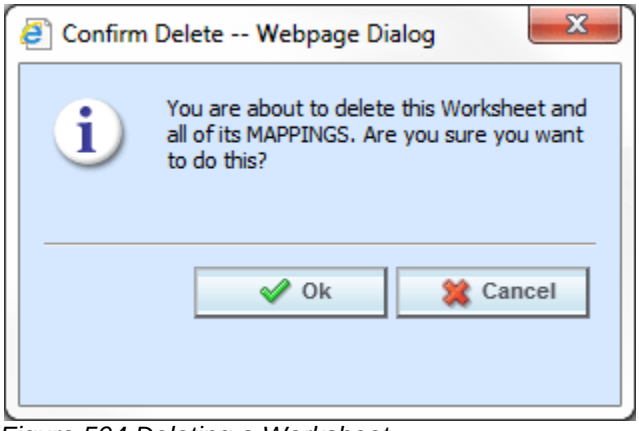


Figure 504 Deleting a Worksheet

1. Select the worksheet you want to delete.
2. Right click and select Delete. The Delete Worksheet screen is displayed.
3. Click OK to delete the worksheet. Click Cancel to return to the Output Mapping tab.
4. Deletes are immediate. The worksheet is removed from the Output Mapping tab and any mappings are removed.

NOTE: *Make sure this is the action you want to take. Deletes cannot be undone.*

MAPPED WORKSHEETS

A mapped worksheet is a worksheet that has been assigned to an output group. Output groups can only have one worksheet mapped. The same worksheet can be mapped to several output groups. The mapped worksheet can be changed or removed at any time.

A worksheet must be mapped to an output group to be visible in Testing, Impact Analysis, or SoftRater Test Interface. Also for Test Case Editor if allowed. If you check to add worksheet and the returned worksheet is blank, check to see if a worksheet has been mapped to the output group.

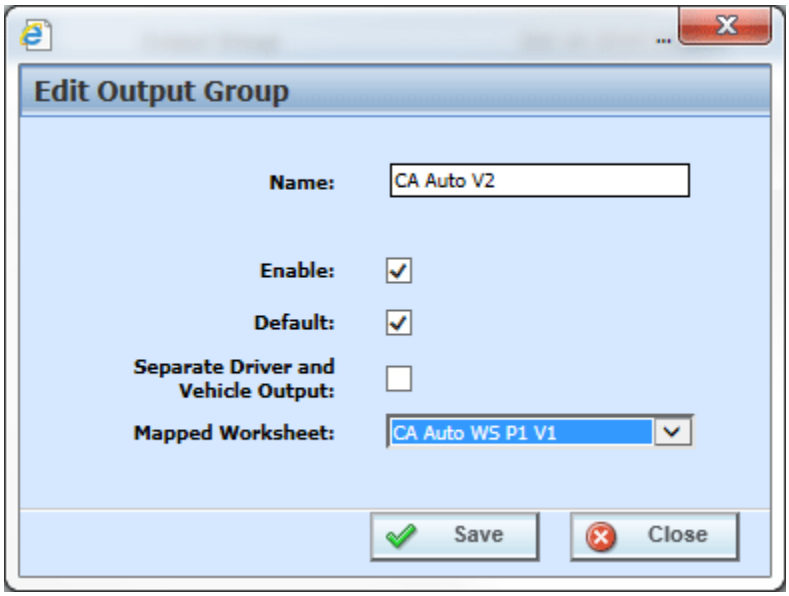


Figure 505 Selecting the Worksheet for the Output Group

1. Select the output group where you want to map a worksheet.
2. Right click and select Options. This opens the Edit Output Group screen.
3. Select the worksheet you want to use.
4. Click Save. The screen closes and the mapped worksheet is listed on the Output mapping tab.

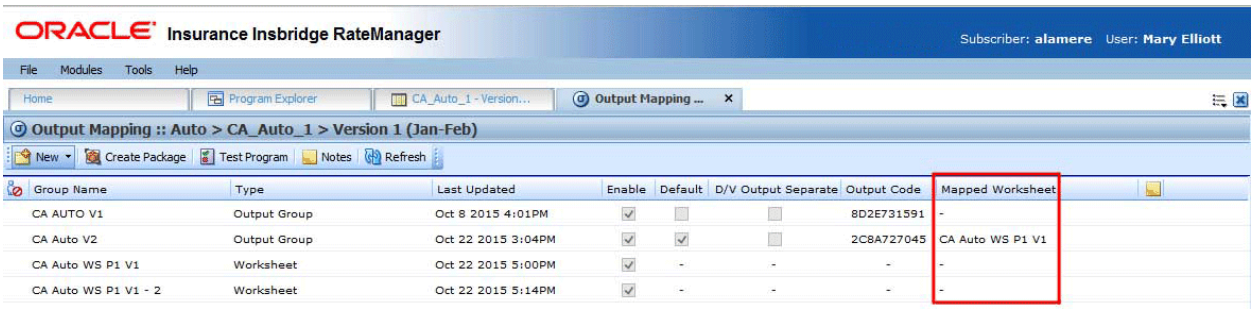


Figure 506 Mapped Worksheet to an Output Group

The worksheet is created every time the program is rated. If you do not need a worksheet, you have to disable the worksheet or delete the worksheet.

Enabling Worksheets in Test Case Editor

If you want a worksheet available in Test Case Editor, it must be activated by a system administrator in the Preferences area. The default is for the worksheet option to be set to false. Most of the time, worksheets are not needed or required when testing a program under development.

The Enable Worksheets in Test Case Editor option governs whether or not a worksheet is returned with the result group in Test Case Editor only. If the Enable Worksheet for Test Case is set to true, then you will have the option to view the worksheet in Test Case Editor. If the option is set to false, the Worksheet XML option will not be available as a View.

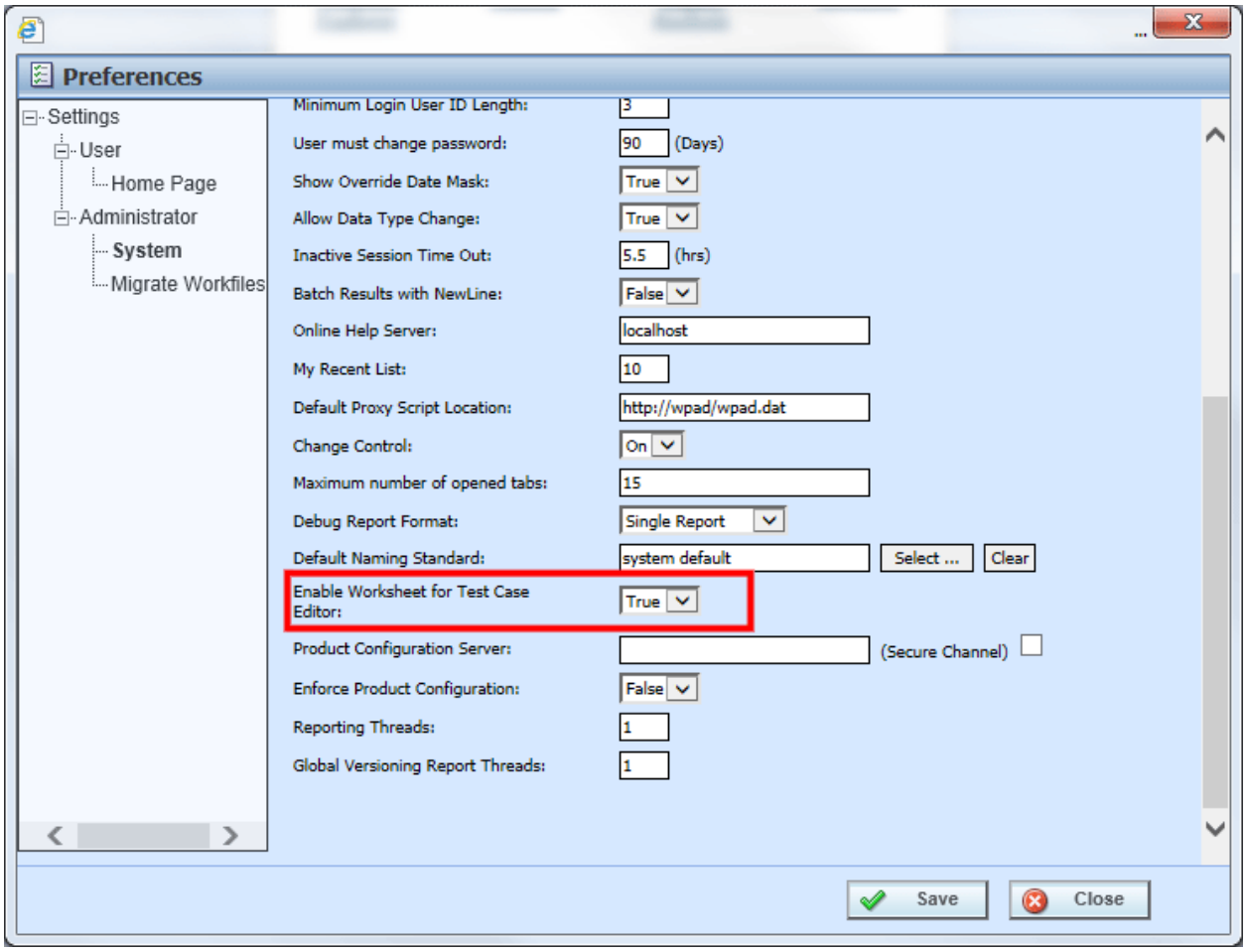


Figure 507 Enabling Worksheets in Preferences

Setting the Enable Worksheet for Test Case Editor to false does not affect the availability or functionality of worksheets in any other area. You can still create, edit, and delete worksheets in Output Mapping. Also, if enabled, the worksheet can be presented in Testing, and Impact Analysis if the Worksheet option is selected. When the program is packaged, the worksheet will be included.

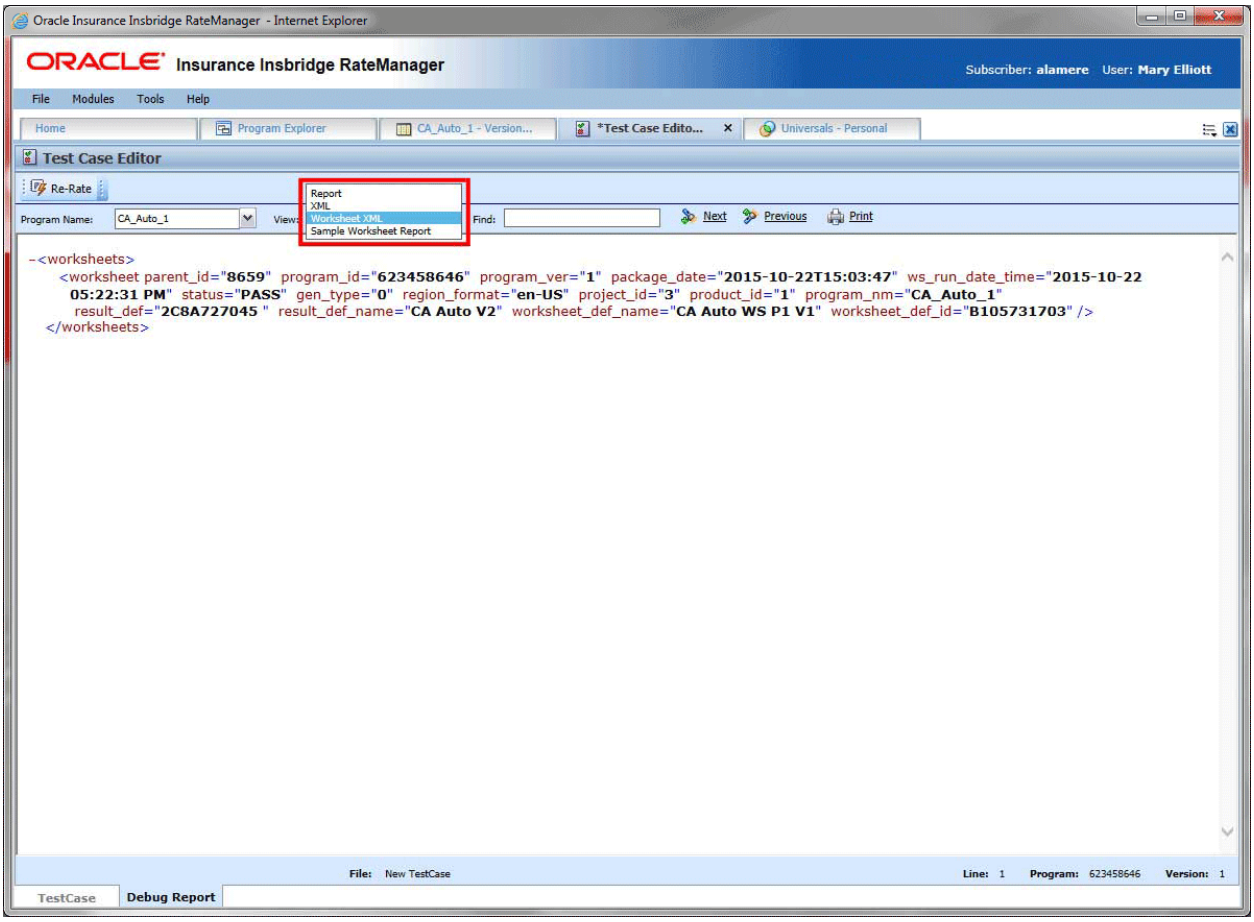


Figure 508 Worksheet Option in Test Case Editor

If a program contains a P2P callout program, the worksheet will show all the executed program worksheets if the mappings are enabled.

Worksheet in Where Used

Worksheet usage is displayed in the View Where Used option. For example, if a calculated variable is included in a worksheet, it will be listed in the Where Used report. Being listed in the Where Used report does not stop a worksheet from being deleted.

Custom XML

You can use any enabled worksheet at runtime by providing the “worksheet_def_name” attribute to the rate request.

Chapter 27

INPUT AND OUTPUT MAPPING

SoftRater requires that input files be in a specific XML format, known as Insbridge.XML. Likewise, when SoftRater finishes rating, it returns the results in Insbridge.XML format as well. An example of an input file in the Insbridge.XML format is shown below:

```
<rate project_ID="5">
    <heading>
        <program parent_id="123" program_id="3" program_ver="1"/>
    </heading>
    <c i="0" desc="Policy">
        <m i="2" n="EffectiveDate" v="04/11/2016"/>
        <m i="4" n="TerritoryCode" v="1"/>
    </c>
    <c i="1" desc="Dwelling">
        <m i="6" n="PerilDeductible" v="250"/>
        <m i="11" n="FireAlarmInd" v="1"/>
        <m i="12" n="BurgularAlarmInd" v="1"/>
        <m i="13" n="OccupancyCode" v="1"/>
        <m i="19" n="DwellingLimit" v="150000"/>
        <c i="2" desc="Coverage">
            <m i="5" n="Coverage">
            <m i="16" n="CoverageCode" v="CovA"/>
            <m i="17" n="CovLimit" v="150000"/>
        </c>
        <c i="2" desc="Coverage">
            <m i="15" n="CovInput" v="3"/>
            <m i="16" n="CoverageCode" v="CovC"/>
            <m i="17" n="CovLimit" v="75000"/>
        </c>
    </c>
</rate>
```

If your organization has an existing XML format you want to use (such as ACORD® XML), then you can use RateManager to map the inputs from your organization's XML format to the Insbridge.XML format. In addition, you can also map the outputs from the Insbridge.XML format to your organization's XML format. This allows you to rate using an input file in your organization's XML format and receive a result file in your organization's XML format. The process looks like this:

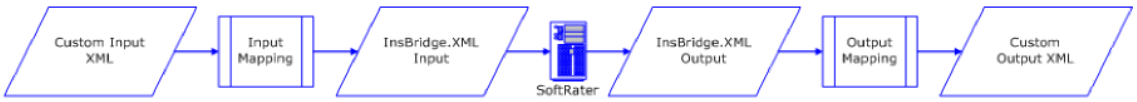


Figure 509 Using an existing XML Format

Alternatively, you can create your own style sheets and have SoftRater convert the input file before rating and convert the output file after rating.

For the purposes of this user guide, custom mappings will refer to any mappings created outside of the Insbridge system.

Preparation for Mapping an Input File

Before you begin the process of mapping, you will need a sample of the custom XML file that will be used during rating. Make sure the file contains all the inputs you want to map. An example of a custom XML file for a fire program is shown below:

```
<Fire>
    <Policy>
        <EffectiveDate>03/05/2016</EffectiveDate>
        <TerritoryCode>1</TerritoryCode>
        <Dwelling>
            <PerilDeductible>250</PerilDeductible>
            <FireAlarmInd>0</FireAlarmInd>
            <BurgularAlarmInd>1</BurgularAlarmInd>
            <OccupancyCode>1</OccupancyCode>
            <DwellingValue>125000</DwellingValue>
            <Coverage>
                <CovCode>CovA</CovCode>
                <Limit>125000</Limit>
            </Coverage>
            <Coverage>
                <CovCode>CovC</CovCode>
                <Limit>125000</Limit>
            </Coverage>
        </Dwelling>
    </Policy>
</Fire>
```

In the above file, the inputs are EffectiveDate, TerritoryCode, PerilDeductible, FireAlarmInd, BurgularAlarmInd, OccupancyCode, DwellingLimit, CovCode and Limit.

MAPPING INPUTS PROCESS

Step 1: Select where you want to map. Inputs can be mapped at either a global level or a program version level.

- If inputs are mapped at a **global level**, then the mapping is available for all programs under that project. This is useful if inputs are the same across all or most of the programs under that project.
- If the inputs are mapped at a **program version level**, then the mapping is available for that program version only.

Step 2: Importing your custom XML. The custom XML must be brought in to RateManager. There are two ways to do this:

- Choose a Custom XML Template from the Library Tab
- Copy and Paste

You can import a mapping from the global level to the program version level or from the program version level to the global level. Once the mapping is in the Custom XML Mapping window, you can map inputs.

Step 3: Mapping Inputs

Regardless of which options you choose, the process of mapping the inputs is the same, with the exception that global mappings are done on the **Program Explorer tab→Projects**, while program version mappings are done from the **Program Explorer tab→Program→Program Version**.

Navigating to Custom XML Mappings at the Global Level

Global mappings include all inputs found in the project.

1. From the Program Explorer tab, select the program version where you want to map inputs.
2. Right click and select **Map Custom XML**.
3. This opens the **Custom XML Mapping** window.

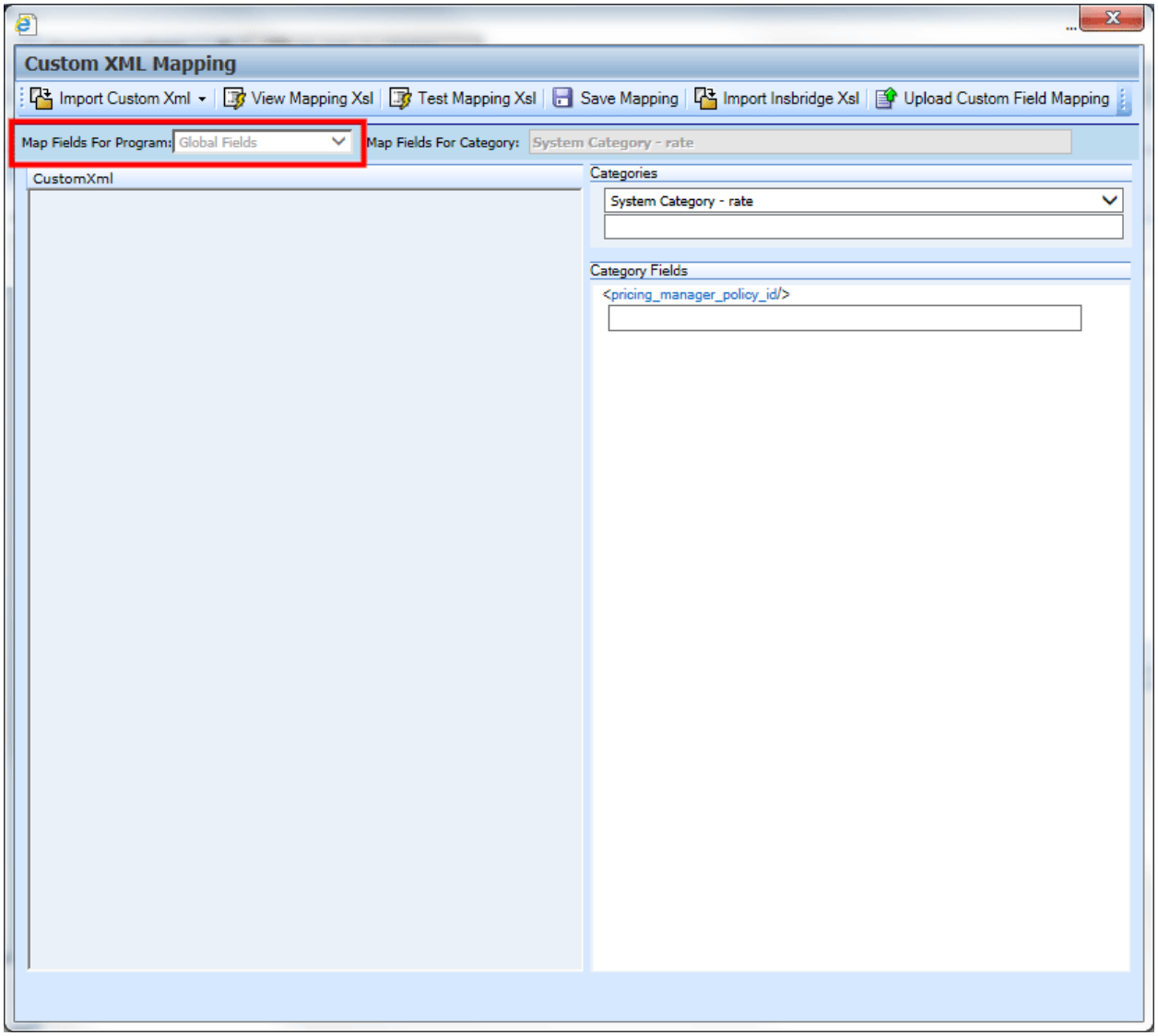


Figure 510 Custom XML Mapping at Global Level

4. At this point, you can import your custom mappings. See **Importing Custom XML** for more information.

You can verify that you are mapping at the global level by looking at the selection in the Map Input For Program field.

Navigating to Custom XML Mappings at the Program Version Level

NOTE: *Before creating a mapping at the program version level, create a new RateManager package for it. You will only be able to include the inputs that were used when the last package was created in your mapping.*

1. From the **Program Explorer** tab, select the **program** and then the **program version** where you want to map inputs.
2. Right click and select **Map Custom XML** from the menu.
3. This opens the **Custom XML Mapping** window.

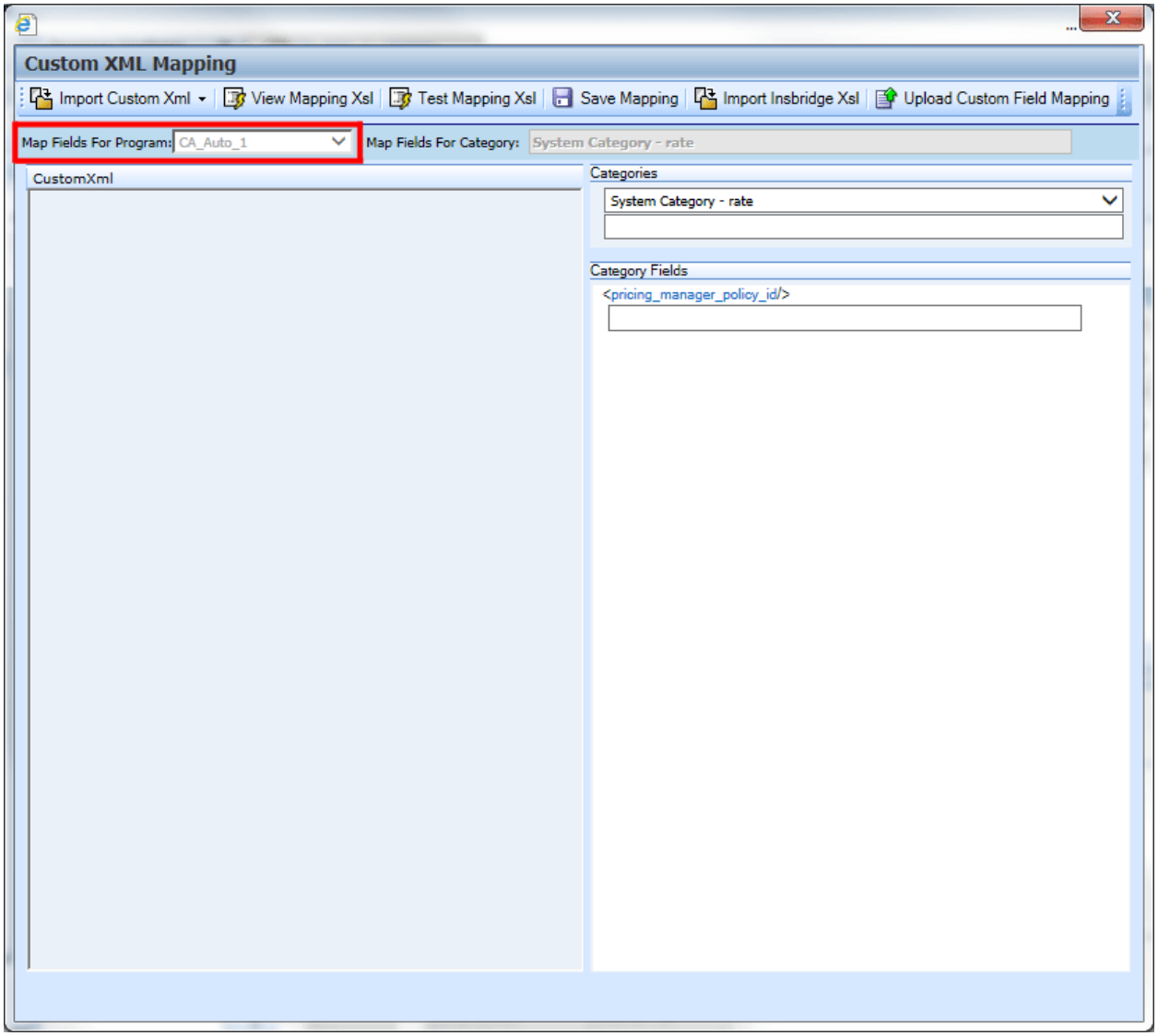
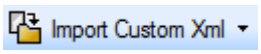
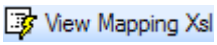
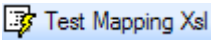
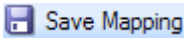
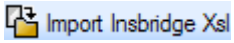
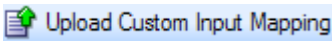


Figure 511 Custom XML Mapping at the Program Version Level

4. At this point, you can import your custom mappings. See the next section, **Importing Custom XML**, for more information.

Custom XML Mapping Functionality

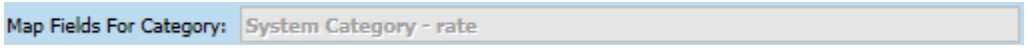
The Menu Bar is located at the top of the Custom XML Mapping window. The menu bar contains the functions you can perform on this window.

	**Import Custom XML:** Has two options: • **Choose Template:** Opens a window where you can select a custom XML template that has been uploaded in to the Library tab. • **Copy/Paste:** Opens a window where you can paste in your custom XML.
	**View Mapping XSL:** Opens a window where you can view, copy and print your input file in XSL format.
	**Test Mapping XSL:** Your custom input file is formatted to Insbridge XML.
	**Save Mapping:** Saves your input file to the local server. Files saved here are available on the Testing module.
	**Import Insbridge XSL:** Opens a saved Insbridge Xsl file.
	**Upload Custom Input Mapping:** Uploads your custom mapping. Uploaded mappings are available in the Testing module.

Custom XML Mapping Screen Details

An information bar is located near the top of the screen. There are two options in the information bar:

- The Map Fields for Program displays the program or global information.



- Map Fields for Category displays the category that you have currently selected.

Custom XML Mapping Work Areas

There are two work areas located in the lower portion of the window.

- **Custom XML**
- **Category Inputs**

Custom XML

The custom XML work space contains your imported custom XML file. You cannot edit your custom XML file here. If you make changes, please return to your XML editing program, make your changes and re-import.

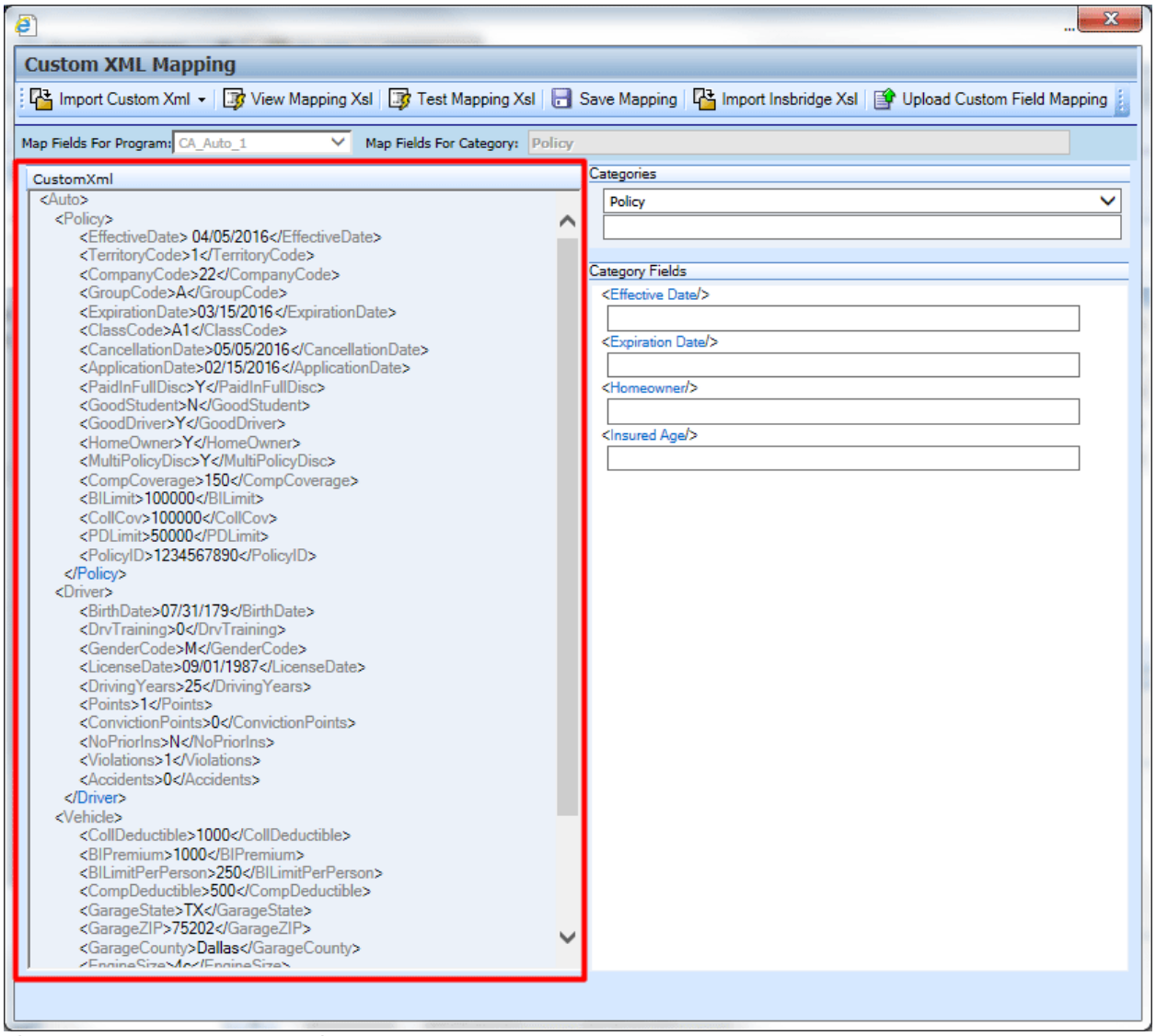


Figure 512 Custom XML Work Area

Category Inputs

The Categories area contains the categories and fields used in your program. You will not be able to edit any of these fields. If you make changes, please return to your program, the globals tab, or the universals tab and make your changes. Package your program prior to mapping your custom XML to ensure that your changes have been incorporated. You do not have to package for a global mapping. Map the category first and then the fields.

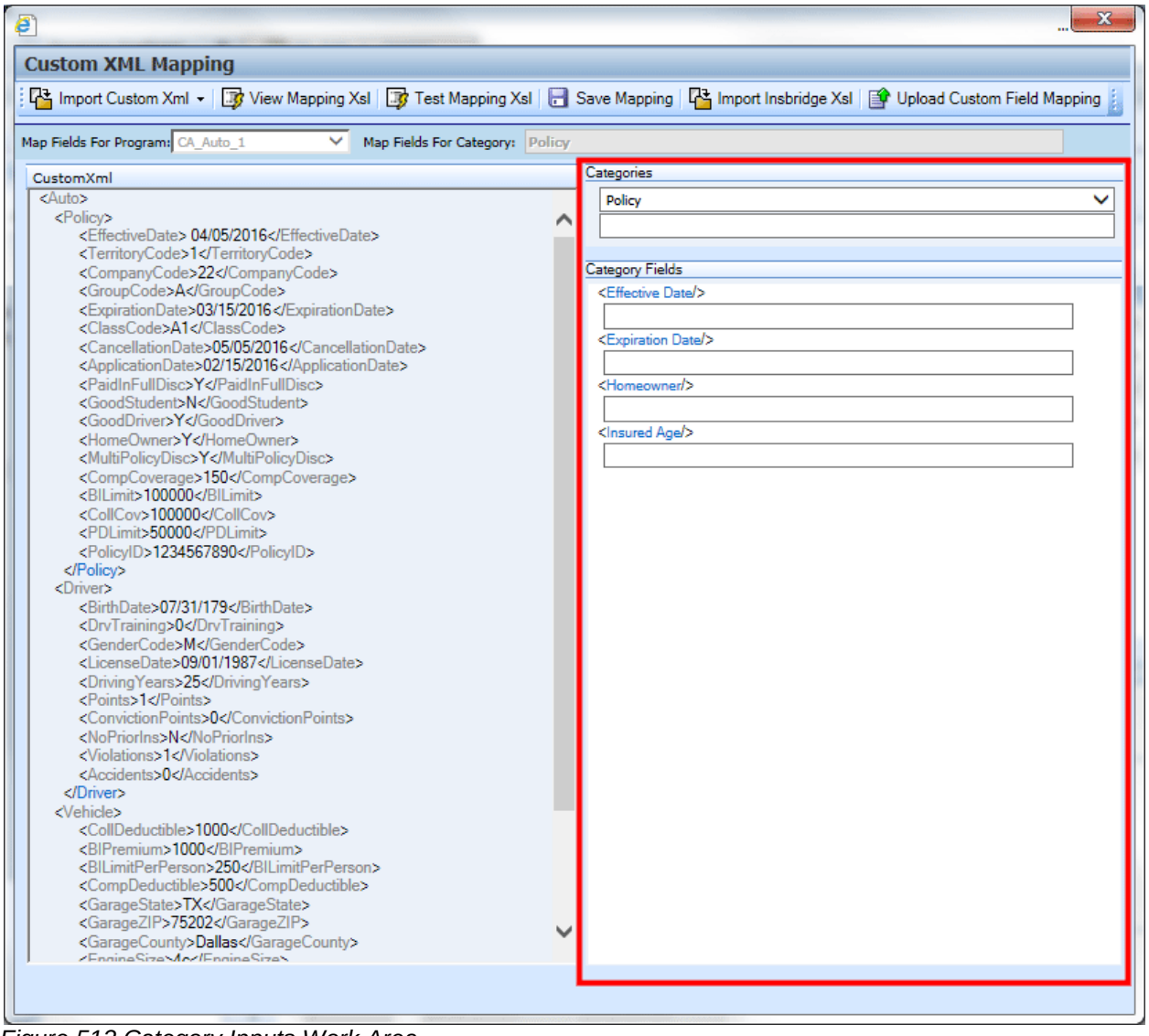


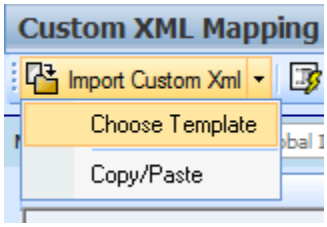
Figure 513 Category Inputs Work Area

The Categories area has a category selection at the top. Select your category and the fields will be populated underneath. At the top level is a system category with one field. All other categories contain the fields used in the program.

NOTE: *When using custom XML in Impact Analysis, you must map the System Category: rate and category input pricing_manager_policy_id. This input requires a unique policy identifier and is used in the batch and pricing manager process.*

Importing Custom XML Using Choose Template

1. From the Input Mapping screen, click **Import Custom XML** and select **Choose Template**.



2. A dialogue box is displayed where you can select the custom XML template you want to use. The selections here have been uploaded to the Custom XML page on the Library tab. If the file you want to use is not listed, please check the Custom XML page on the Library tab.

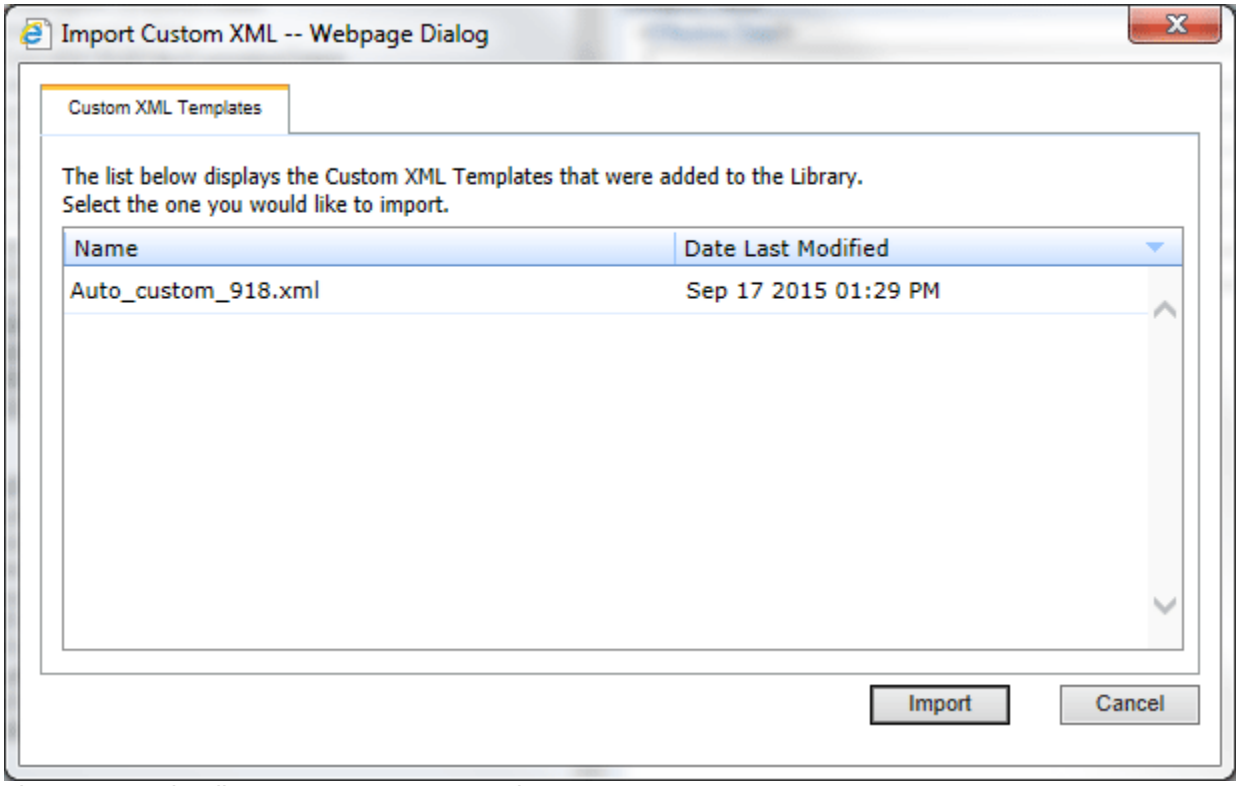


Figure 514 Uploading a Custom XML Template

3. Click Import.

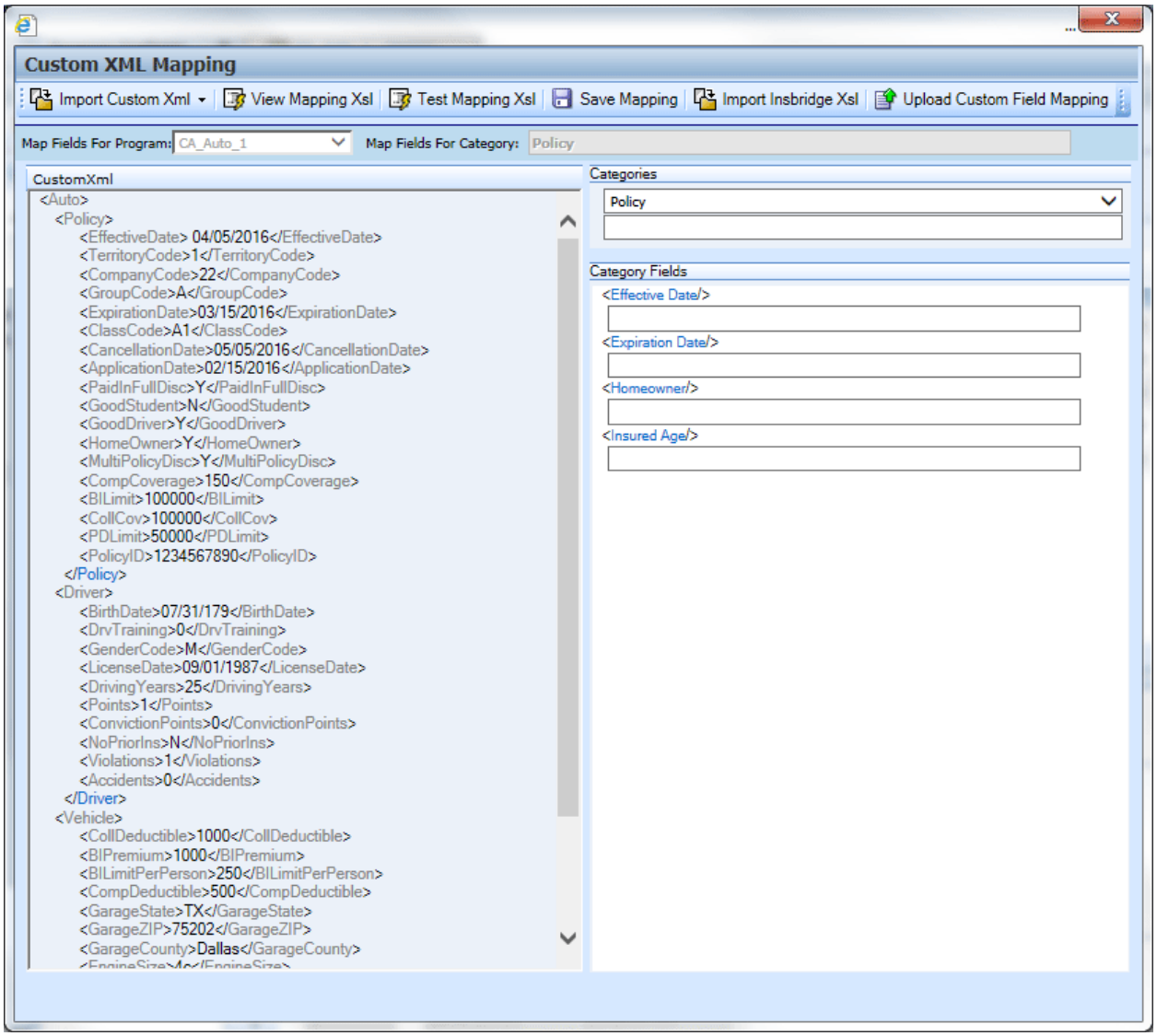


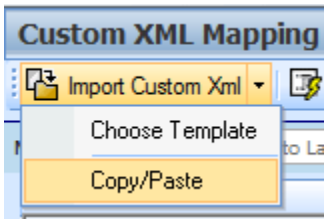
Figure 515 Imported Custom XML Template

4. The custom XML is imported into RateManager and the **Custom XML Mapping** screen refreshes with the selection.

The file is ready to be mapped. Please see Creating a New Input Mapping.

Importing Custom XML Using Copy/Paste

1. From the Input Mapping screen, click **Import Custom XML** and select **Copy/Paste**.



2. A text box is displayed where you can paste your custom XML file.

3. Outside of RateManager, open your custom XML file in a text editor or XML editor. Notepad is the simplest and fastest method however; you can use any XML editor you want.

4. In Notepad, select all of the text by selecting **Edit→Select All** or by pressing **Ctrl+A**.

5. Copy the text to the clipboard by selecting **Edit→Copy** or by pressing **Ctrl+C**.

6. Return to the text box in RateManager and paste the text by right clicking the text area and selecting **Paste** from the menu or by pressing **Ctrl+V**.

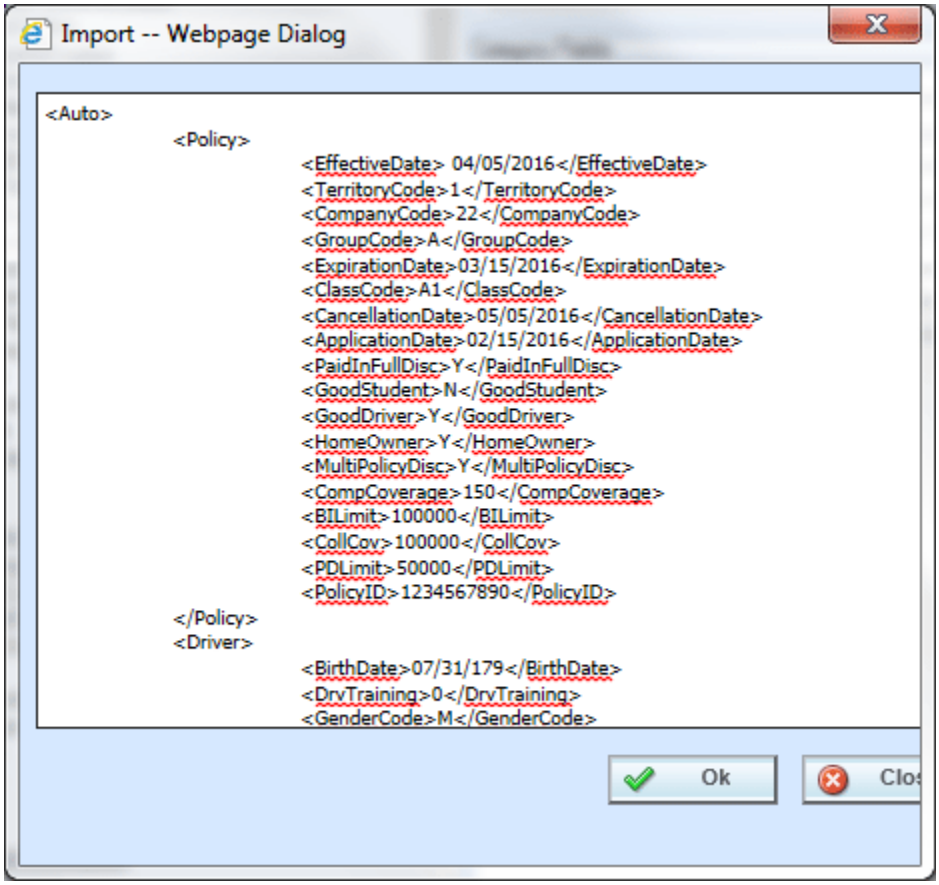


Figure 516 Import Custom XML Text Box

7. Click **OK**.

8. The custom XML is imported into RateManager and the **Custom XML Mapping** screen will refresh with the selection.

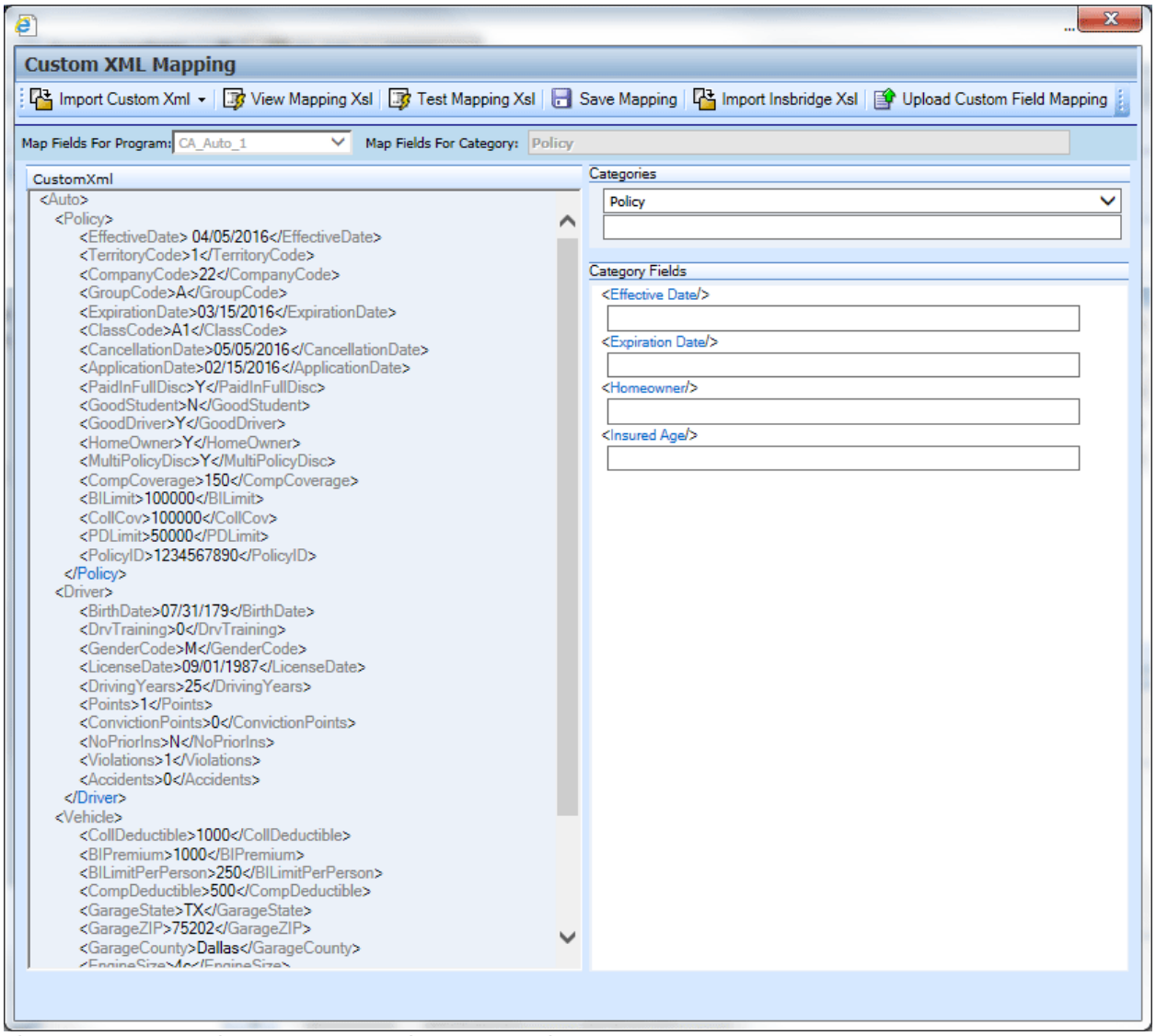


Figure 517 Imported Custom XML Using Copy and Paste

Creating a New Input Mapping

Regardless of which type of input mapping you are creating or how you brought in your custom XML, the steps for completion are the same. You must be on the Custom XML Mapping screen before you can proceed. You can verify that you are mapping either at the global level or at the program version level by looking at the selection in the Map Inputs For Program field.

9. Now that your custom XML is in RateManager, you have to tell RateManager which category in the custom XML file goes with which category in RateManager. This is done by dragging and dropping the category from the CustomXml area on the left to the blank text box in the Categories

area on the right.

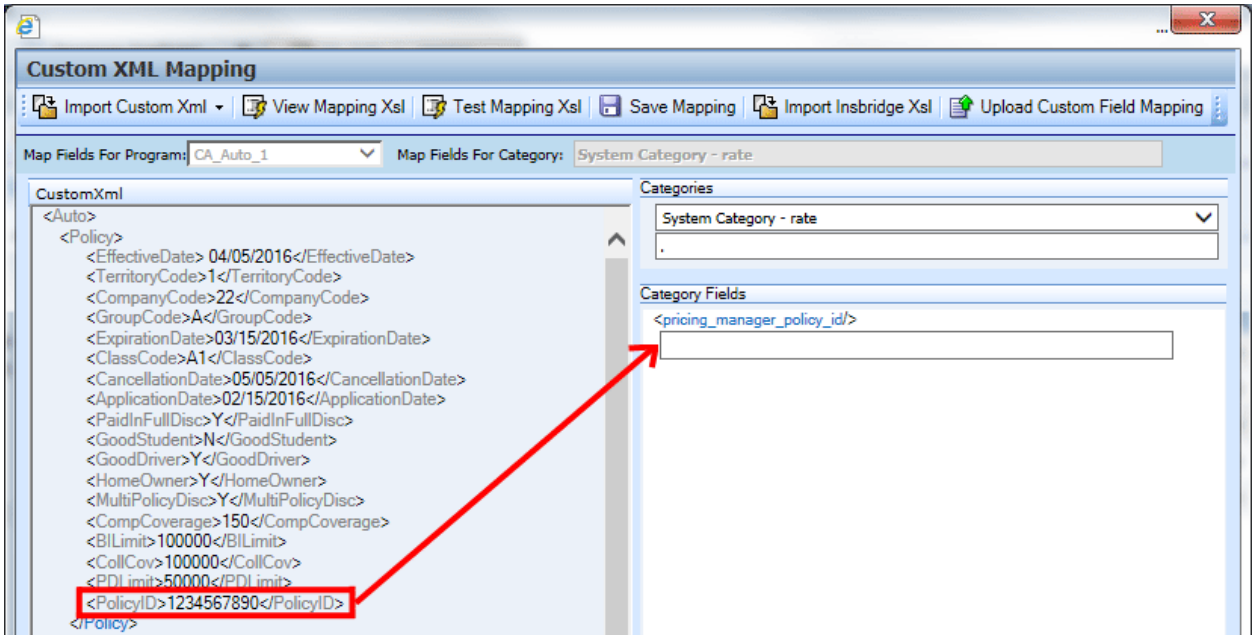


Figure 518 Mapping Custom XML

10. If you are going to use Impact Analysis, you must map the System Category: rate. The only input in this category is pricing_manager_policy_id. System Category: rate can be mapped with a single dot to indicate a top level category. You will not need to select a category from the CustomXml area. For the category input, use a unique identifier such as Policy Number. Highlight the input you want. Drag your cursor over to the field. A plus sign will be displayed when you reach the field. Release your cursor and the field will be filled with the input you have selected.

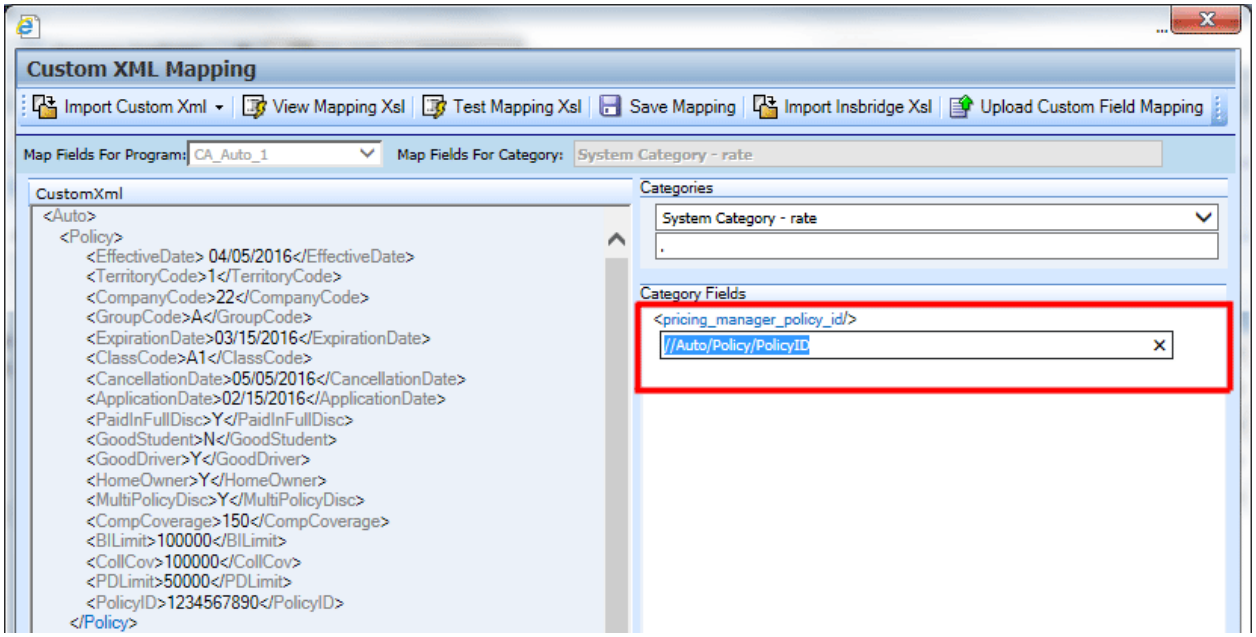


Figure 519 Mapping System Category: rate

11. Make sure you map all of your RateManager categories before mapping the inputs. To switch between categories, use the menu located at the top of the Categories. If you are mapping at the

program version level, you will only see the categories that are used in the program. If you are mapping at the global level, you will see all categories. To map a category, highlight the category you want in the CustomXML area. Drag your cursor over to the field underneath your selected category in the Categories area. A plus sign will be displayed when you reach the field. Release your cursor and the field will be filled with the category you have selected.

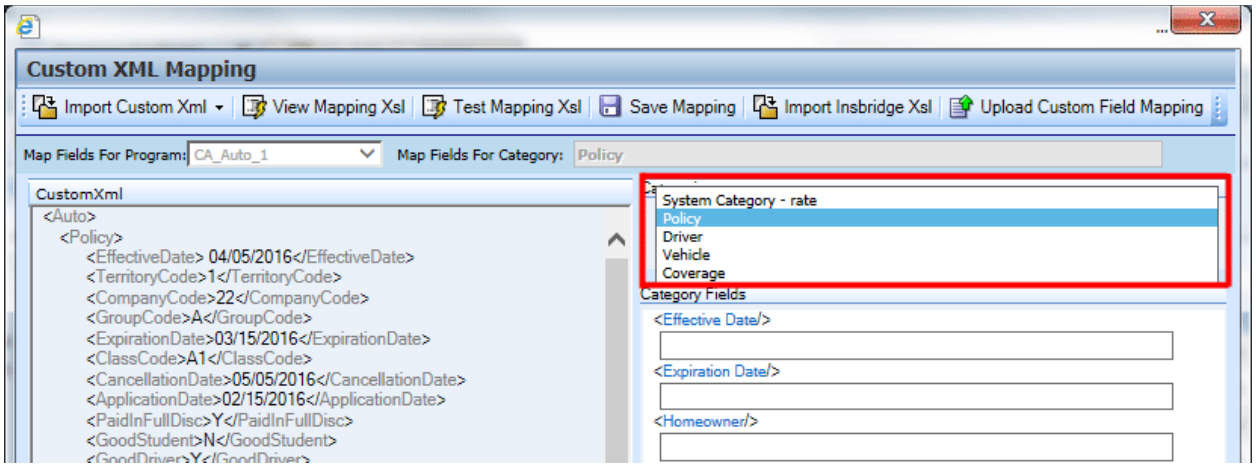


Figure 520 Selecting Categories

12. Once you have mapped all the categories, you can begin mapping the inputs. Inputs are mapped using the same drag and drop technique used to map the categories. If you are mapping at the program version level, you will only see the inputs used by the current program. Each Input located under the Category Inputs area needs to have a value. Be sure to map all the inputs. If you are mapping at the global level, you will see all inputs for the current . You will only need to map the inputs that are used in the programs that will use the mapping.

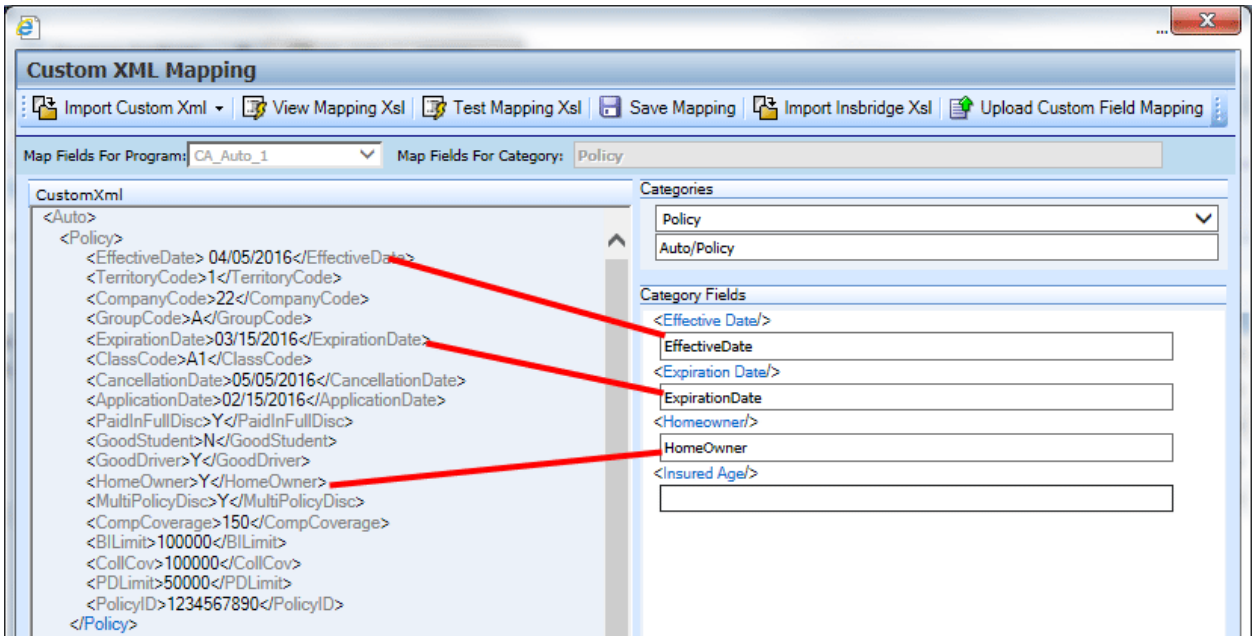


Figure 521 Selecting Inputs

13. After you have finished mapping the categories and inputs, click **Save** to save the mapping.

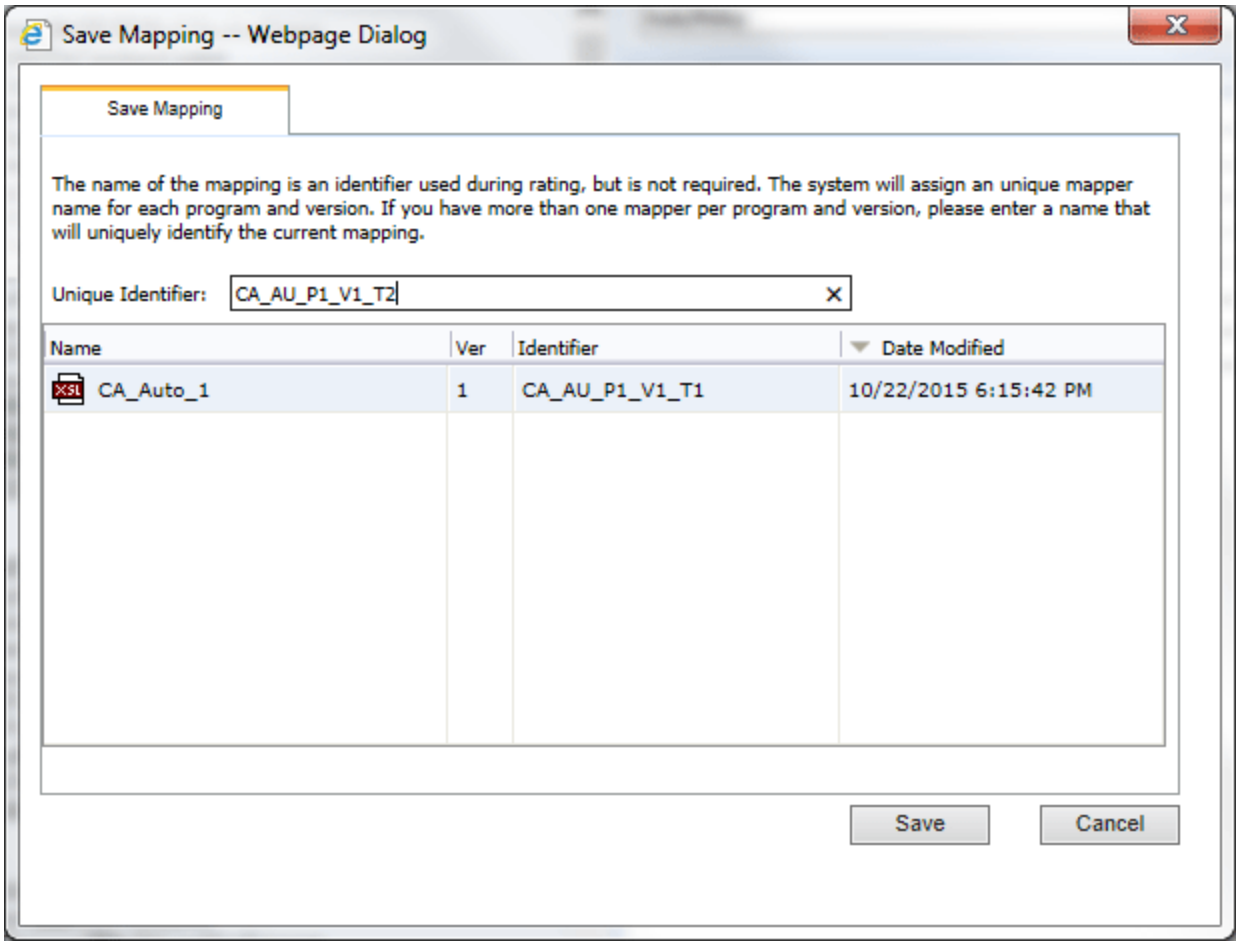


Figure 522 Save Mapping

14. Click **Save** to save your mapping.

Save Mapping

The save mapping window is where you can specify a unique identifier for the mapping. Unique identifiers allow you create multiple versions. You can create as many versions as you need. Versions are listed in the space underneath. If there is no identifier, then the default identifier has been used.

Saves your input file to the local server. Files saved here are available on the Testing module.

When Saving a Mapping

1. Enter a unique identifier. If you do not enter a unique identifier, the mapping is saved using your subscriber ID, the project ID, the program ID and the program version. For example, 123_1_6_1.xml, which means subscriber 123, project Alamere (1), program 6, version 1. The screen defaults to the last unique identifier or default identifier name used. If you save without changing the name, you will write over the previous version. No warning is issued.

For a global mapping, the mapping is saved using only your subscriber ID and the project number. For example, 123_1.xml, which means subscriber 123, project Alamere (1).

If you do enter a unique identifier, the mapping is saved using the same format, except the filename will be appended with the unique identifier you entered. For example, if you saved the mapping as InputMapping, then it would be saved as 123_1_6_1.InputMapping.xml at the program version level or 123_1.InputMapping.xml at the global level.

Although you do not need the filename when rating, you will need to know your subscriber ID, the project number, the program ID, the program version and the unique identifier for the mapping.

Mappings created from templates have the template name for the unique identifier.

2. Finally, you will need to create a new package for the program you created the mapping for. If you created a global mapping, you will need to create a new package for every program that will use the mapping.

3. For information on using the mapping while rating, see Rating a File with a Mapping.

If you are batch rating, you can select the input mapping. There are three options; Global, Local and Custom. Global uses XSLT created for the product (all inputs for the product), local uses XSLT created for a specific version (only inputs used by a specific version) and custom uses a custom XSLT created outside of the RateManager mapping tool.

Save Mapping files are saved on the server where the Insbridge system is installed. To verify files, you must access to the Insbridge server. Files are located:

C:\Program Files\Oracle\IGBU\Insbridge\Webs\RM4\Workfiles\Mappings\Input

Save Mappings Options

Saved mappings has a right click menu with two options:

- **Preview Mapping**: Displays the mapping in the View Mapping XSL window. No changes are allowed.

- **Delete**: Deletes the selected mapping. A warning message is displayed before deleting.

View Mapping Xsl

The View Mapping Xsl option is where you can view, copy, and print your input file in XSL format. This is a view only window. No edits are allowed. If you want to edit, you must copy and edit in an outside editing program.

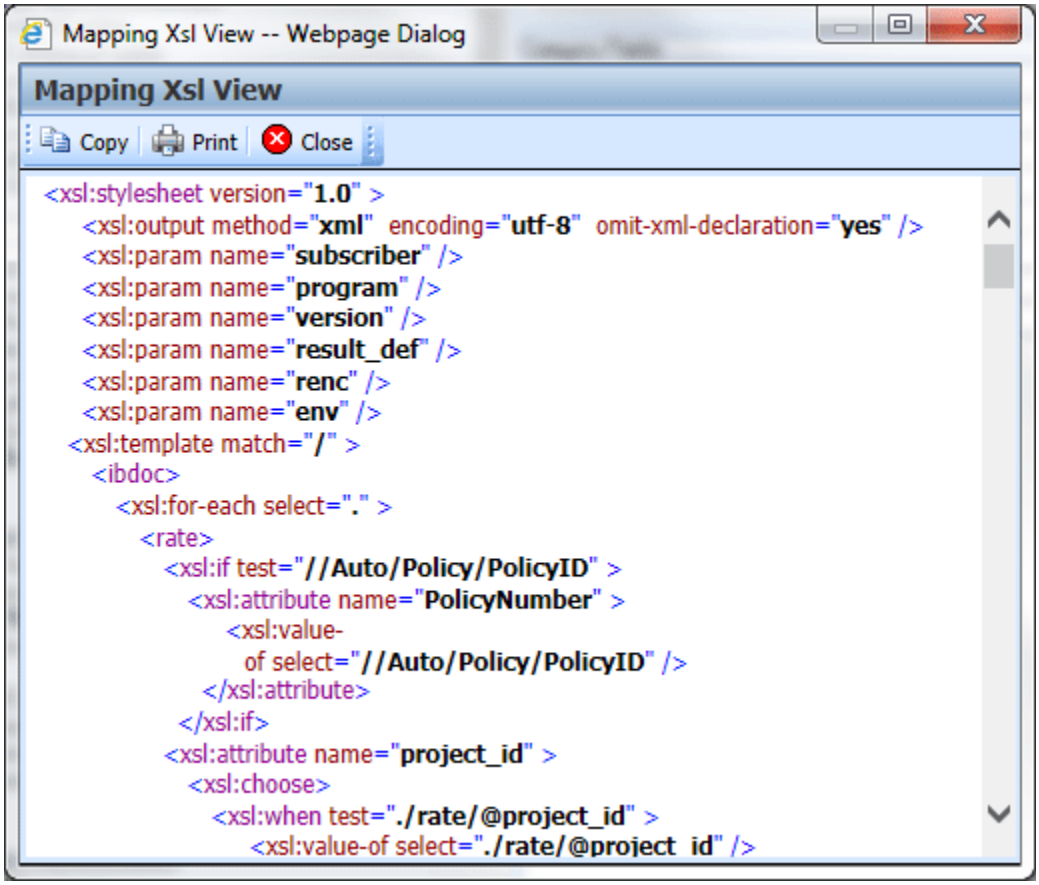


Figure 523 View Mappings Xsl

There are three buttons in the top bar menu:

- **Copy:** copies the Xsl to a clipboard. Use this when you want to edit the file outside of RateManager.
- **Print:** prints the contents of the window to a local or network printer.
- **Close:** closes the window.

Test Mapping Xsl

The Test Mapping Xsl option converts your file in to Insbridge XML. On this window you can view, copy, and print your input file in Insbridge XML format. This is a view only window. No edits are allowed. If you want to edit, you must copy and edit in an outside editing program.

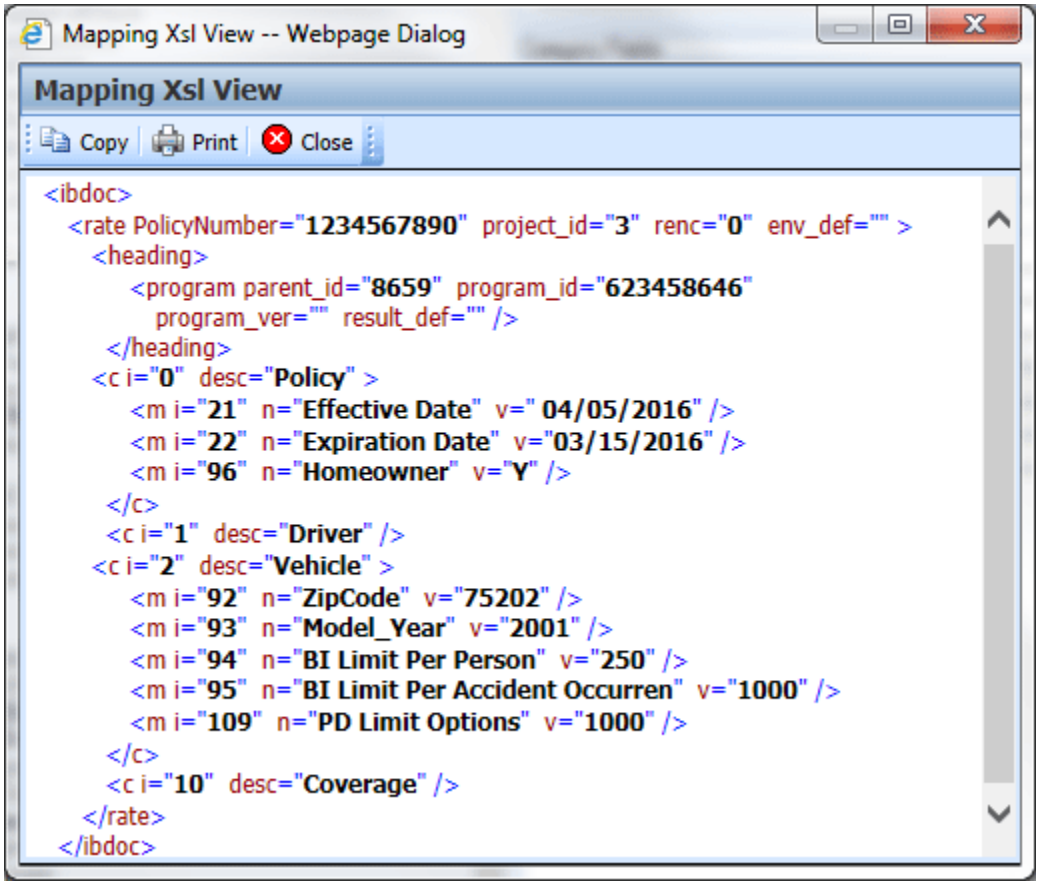


Figure 524 Test Mapping Xsl

There are three buttons in the top bar menu:

- **Copy:** copies the Xsl to a clipboard. Use this when you want to edit the file outside of RateManager.
- **Print:** prints the contents of the window to a local or network printer.
- **Close:** closes the window.

Import Insbridge Xsl

The Import Insbridge Xsl option is where you can import a program version or global custom mapping that has been saved to the Insbridge server. This screen also will allow you to view or delete any custom mapping file that has been loaded to the Insbridge server. Clicking on the Import Insbridge Xsl option opens a separate window where you can select the type of mapping file you want.

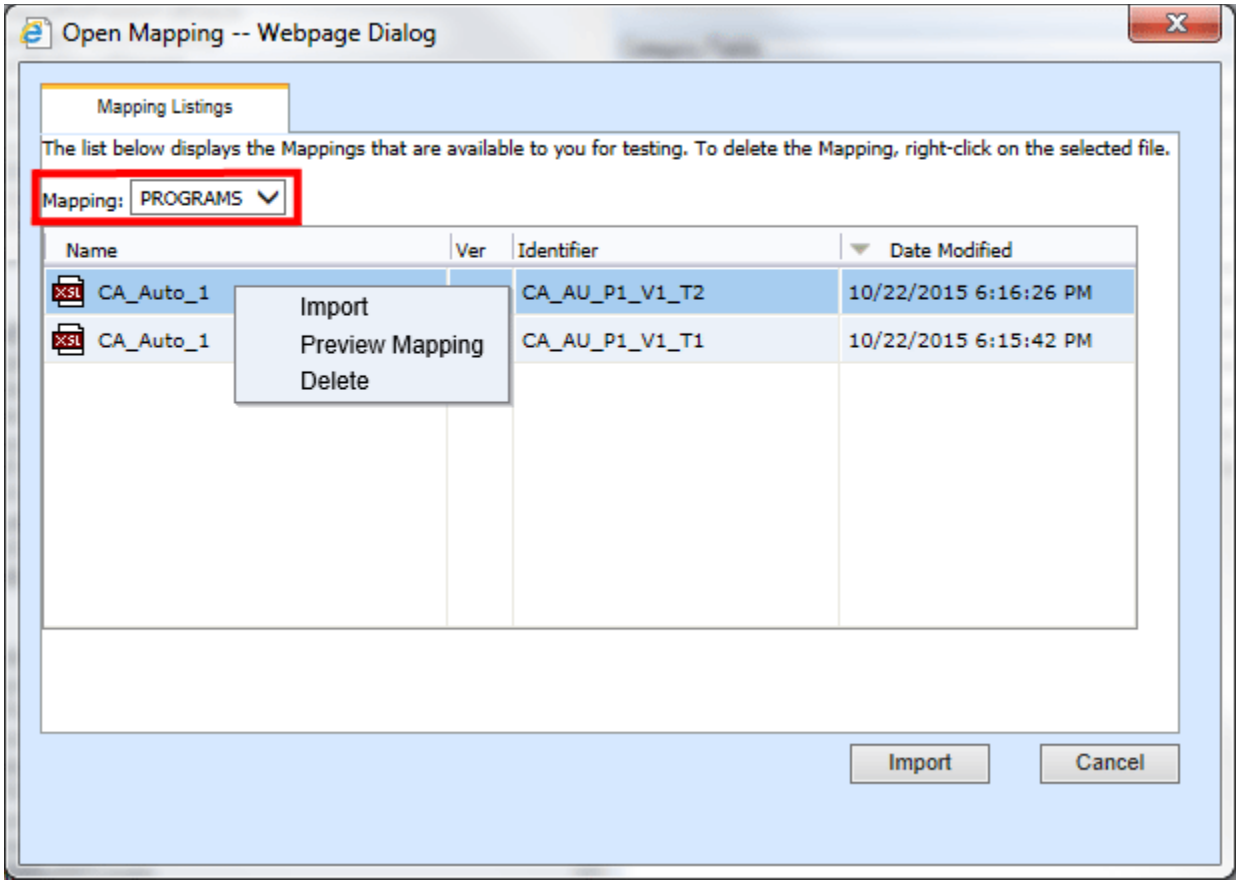


Figure 525 Open Mapping Right Click Menu

1. Select the type of mapping you want from the Mapping Listing. There are three options:

    - **PROGRAMS:** A mapping that is specific to a particular program version. Custom program version mappings are created on the **Program**.
    - **GLOBALS:** A mapping that can be used by all programs under a product. Custom global mappings are created on the project.
    - **CUSTOM:** A mapping that was created outside of RateManager and loaded into the Insbridge system. A custom mapping can be viewed or deleted on the Open Mapping screen. You will not be able to Import.

    The available files will be displayed in the area below. If the Import option is grayed out, that file is not available for import or you do not have the proper permissions.

2. Select the file you want.

3. Click **Import**. The mapping screen will close and your mapping file will be loaded in to the Custom Xml area on the Mapping screen.

4. If you want to view or delete a custom mapping, right click and select the option you want.

NOTE: *Import Custom Mapping usage depends upon your group rights. If your group has the rights to write to a program, you will have the right to import mappings. If the Import option is not listed, your group does not have the right to import.*

Right Click Menu

Every mapping file has a right click menu available.

- **Import:** Imports the selected file into the Custom Xml area on the Mapping screen.
- **Preview Mapping**: Displays the mapping in the View Mapping XSL window. No changes will be allowed.
- **Delete**: Deletes the selected mapping. A warning message will be displayed before deleting.

Custom Mapping Files

Custom Mappings are any custom mapping files that have been loaded to the Insbridge server. Any custom mapping file that has been included in an SRP and loaded to any SoftRater node will be place in a custom mapping location on the server. The custom mapping files that will be displayed will be the files found in this location.

Upload Custom Input Mapping

The Upload Custom Input Mapping option is where you can upload your custom mappings. Uploaded mappings are available in the Testing module. Files must be xslt format.

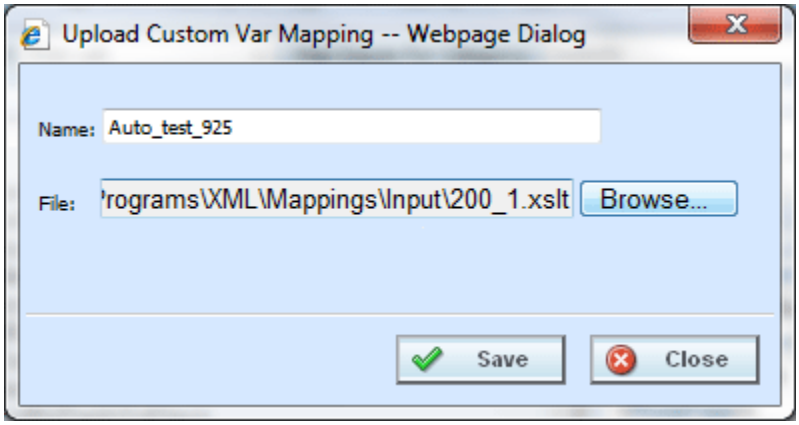


Figure 526 Upload Custom Input Mapping

Uploaded files are saved on the server where the Insbridge system is installed. To verify files, you must access to the Insbridge server. Files are located:

C:\Program Files\Oracle\IGBU\Insbridge\Webs\RM4\Workfiles\Mappings\Input\Custom

CREATING A NEW OUTPUT MAPPING

Unlike an input mapping, an output mapping can only be created for a specific program version (program version level). This is due to the fact that result mappings are created at the program version level.

Preparation for Mapping an Output File

Before you begin the process of creating a mapping, you will need a sample of the custom XML file that you want to receive after rating. Make sure the file contains all the outputs that you will need. An example of a custom output XML file for a fire program is shown below:

```
<Fire>
    <Policy>
        <TotalPolicyPremium>250</TotalPolicyPremium>
        <Dwelling>
            <LocationNumber>1</LocationNumber>
            <Coverage>
                <CovCode>CovA</CovCode>
                <CovPremium>125</CovPremium>
            </Coverage>
            <Coverage>
                <CovCode>CovC</CovCode>
                <CovPremium>125</CovPremium>
            </Coverage>
        </Dwelling>
    </Policy>
</Fire>
```

In the above file, the outputs are TotalPolicyPremium, LocationNumber, CovCode and CovPremium.

Output Mapping Special Characters

If using the custom mapping tool in RateManager for output mapping, these characters cannot be part of an element or attribute name in target output document:

Name	Character
Asterisk	*
At Sign	@
Backslash	\
Colon	:
Comma	,
Dollar Symbol	$
Empty Space	' '

Equal Sign	=
Exclamation Point	!
Greater-Than Sign	>
Less-Than Sign	<
Minus Sign	-
Parenthesis	( )
Period (dot)	.
Plus Sign	+
Question Mark	?
Quotation Mark	"
Semicolon	;
Slash	/
Square Brackets	[ ]
Vertical Bar	|

Mapping Outputs

1. Navigate to the **Output Mapping** tab for the program and version where you want to create a custom output mapping.

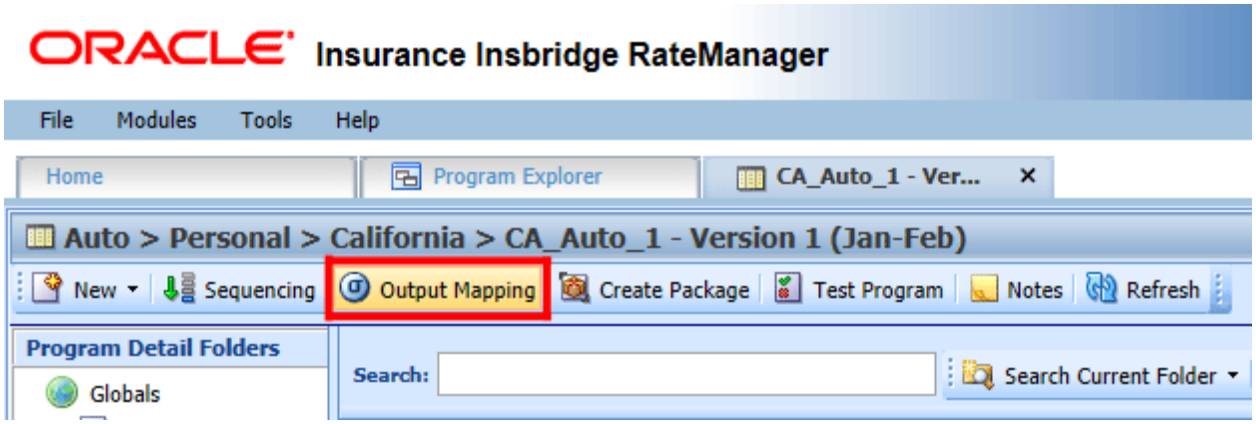


Figure 527 Navigating to Output Mapping

2. On the **Output Mapping Group Listing** screen, double-click the group you want the output mapping based on to open it.

3. This opens the **Edit Group** screen.

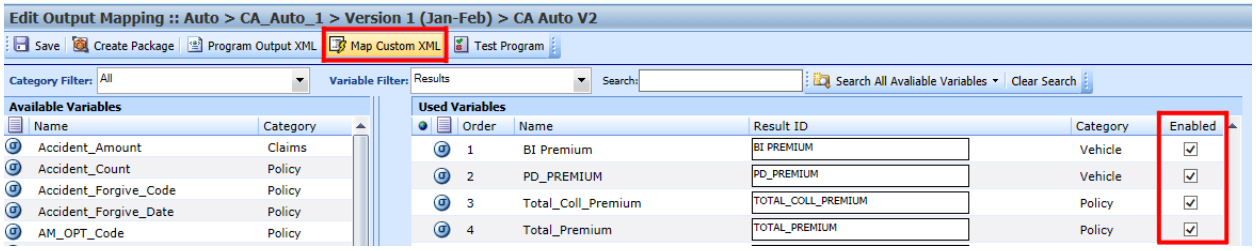


Figure 528 Creating a Custom Output Mapping

4. Check that all the results you want to see in the output have been added to the group. Also make sure there is a check in the **Enabled** box at the end of the row for each result you want to see in the output. If you make any changes, be sure to save the group before creating a mapping.

5. Click **Map Custom XML**. This opens the **Map Custom Output XML** window.

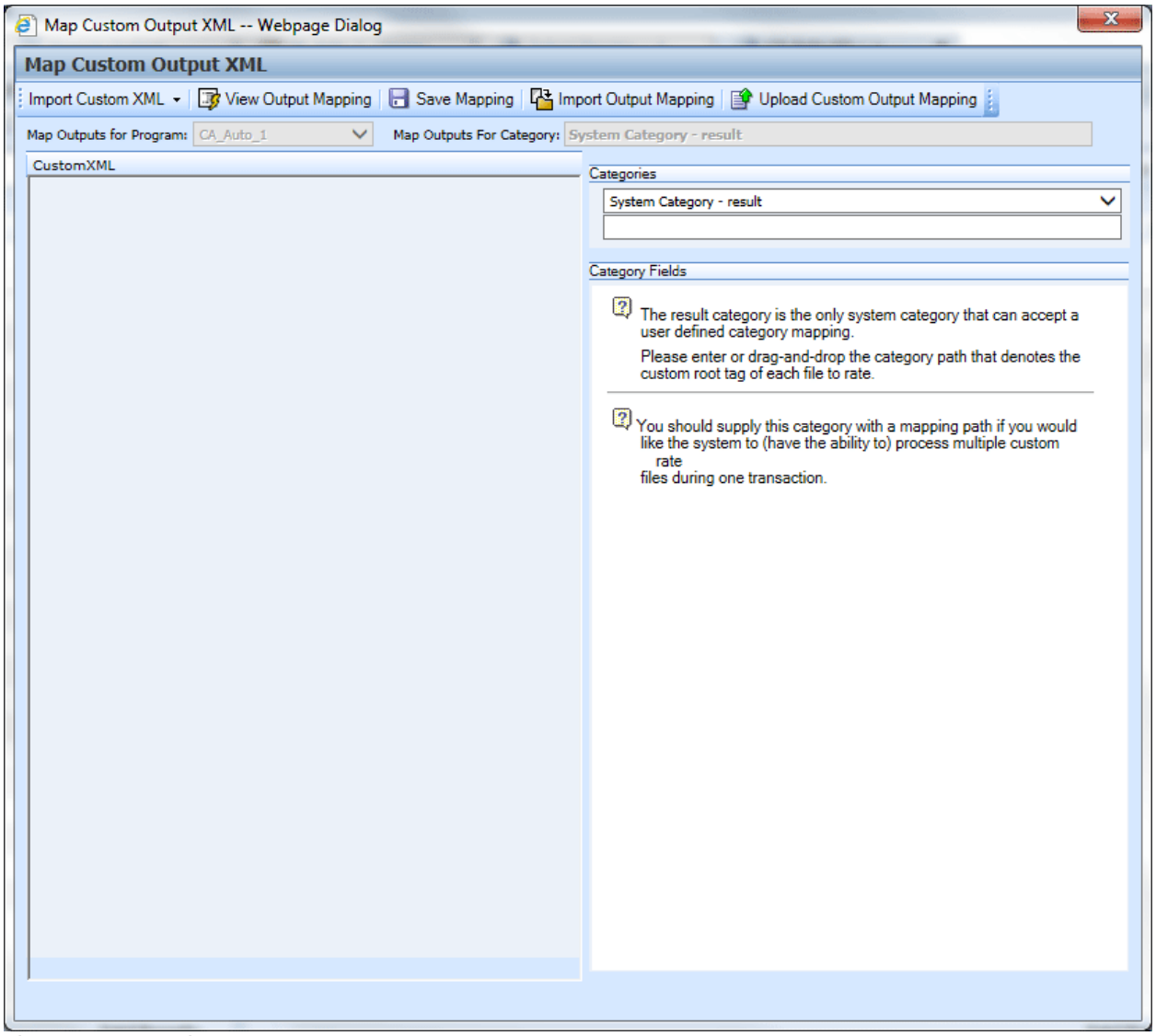
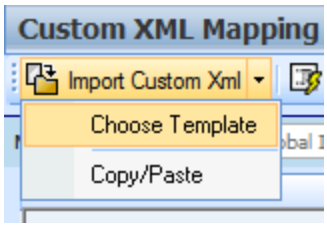


Figure 529 Mapping Custom Output XML

Importing Custom XML Using Choose Template

The importing of custom output XML files is handled in the same manner as custom input files

1. From the Map Custom Output XML window, click **Import Custom XML** and select **Choose Template**.



2. In the dialogue box, select the custom XML template you want to use. The selections here have been uploaded to the Custom XML page on the Library tab. If the file you want to use is not listed, please check the Custom XML page on the Library tab.

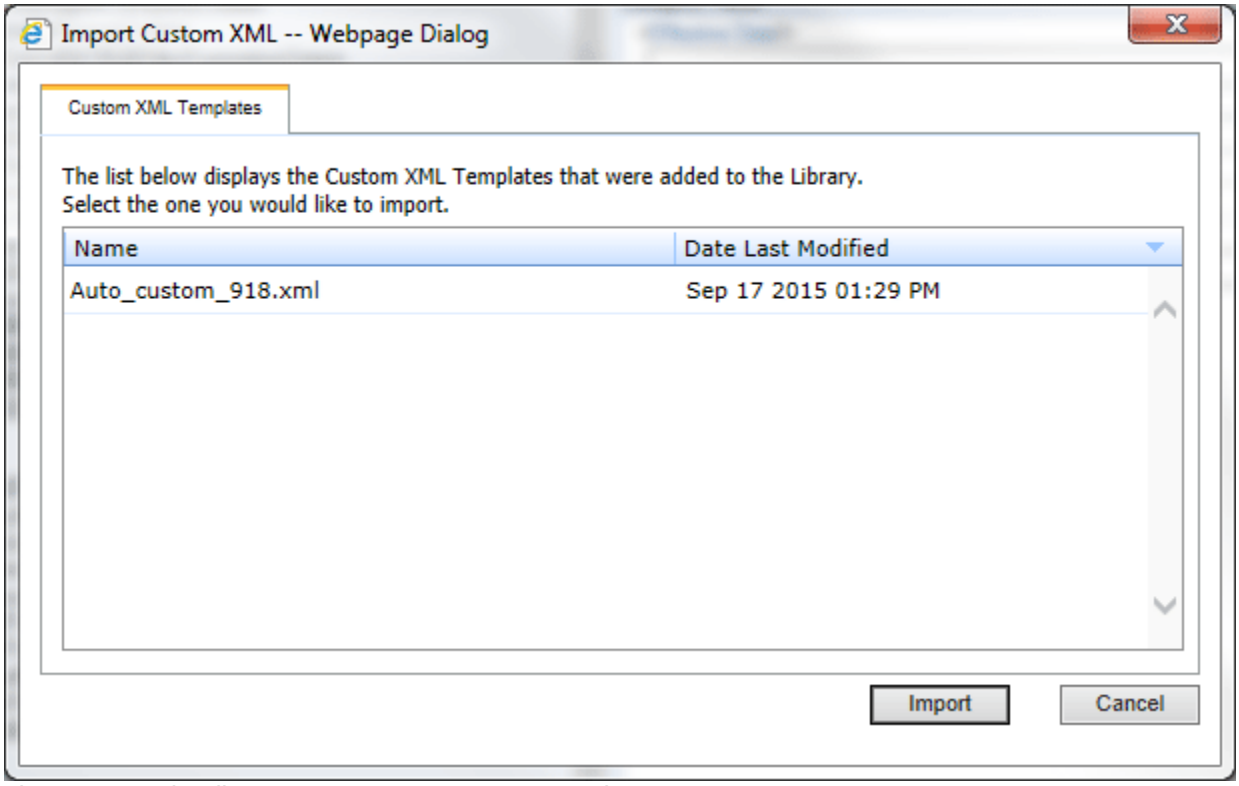


Figure 530 Uploading a Custom Output XML Template

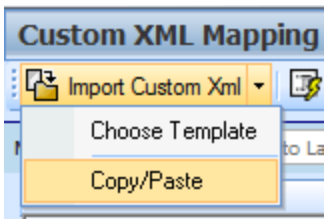
3. Click Import.

4. The custom XML is imported into RateManager and the Map Custom Output XML screen refreshes with the selection.

The file is ready to be mapped. Please see Creating an Output Mapping.

Importing Custom XML Using Copy/Paste

1. Click **Import Custom XML**.



2. A text box where you can paste your custom XML file is displayed.

3. Outside of RateManager, open your custom XML file in a text editor or XML editor. Notepad is the simplest and fastest method however; you can use any XML editor you want.

4. Select all of the text by selecting **Edit➤Select All** or by pressing **Ctrl+A**.

5. Copy the text to the clipboard by selecting **Edit➤Copy** or by pressing **Ctrl+C**.

6. Return to the text box in RateManager and paste the text by right clicking the text area and selecting **Paste** from the menu or by pressing **Ctrl+V**.

7. The XML is pasted into the text box.

8. Click **OK**.

9. The custom XML is imported into RateManager and the **Map Custom Output XML** screen refreshes.

Creating an Output Mapping

Regardless of which type of output mapping you are creating or how you brought in your custom XML, the steps for completion are the same. You must be on the Map Custom Output XML Mapping screen before you can proceed.

1. Now that your custom XML is in RateManager, you have to tell RateManager which category in the custom XML file goes with which category in RateManager. This is done by dragging and dropping the category from the Custom Xml area on the left to the Categories text box on the right.

2. Repeat this process for all of your RateManager categories before proceeding to the outputs. To switch between categories, use the drop-down text box. You will only see the categories that are used in the current program version.

3. Once you have finished the categories, you can work on the outputs. Outputs are done using the same drag and drop technique used for the categories. You will only see the outputs used by the current program version. Each RateManager output on the right needs to have a value.

4. Once you have finished the categories and outputs, click **Save** to save the mapping.

Save Output Mapping

The save mapping screen is where you specify a unique identifier for the mapping. Unique identifiers allow you create multiple versions. You can create as many versions as you need. Versions will be listed in the space underneath. If there's no identifier, then the default identifier has been used.

Right clicking any saved mapping to:

- View the custom mapping
- Delete the mapping. Deleting does not issue a warning message.

5. If you do not enter a unique identifier, the mapping will be saved using your subscriber ID, the project ID, the program ID and the program version. For example, 123_1_6_1.xslt, which means subscriber 123, project Alamere (1), program 6, version 1.

   If you do enter a unique identifier, the mapping will be saved using the same format, except the filename will be appended with the unique identifier you entered. For example, if you saved the mapping as OutputMapping, then it would be saved as 123_1_6_1.OutputMapping.xslt.

   Although you do not need the filename when rating, you will need to know your subscriber ID, the project ID, the program ID, the program version, and the unique identifier for the mapping.

6. Finally, you have to create a new package for the program you created the mapping for.

7. For information on using the mapping while rating, see Rating a File with a Mapping.

Rating a File with a Mapping

To test your mapping, you can use the Testing module or IBFA (Insbridge Framework Administrator). You will only be able to test input mapping in Testing. Output mapping can only be tested in the IBFA. In production, the SoftRater WSI is used to specify the mappings. For more information on this, see the SoftRater topic Sample SOAP Request.

Before starting to test, be sure you have created a package. If you are testing in IBFA, you are most likely using a SoftRater environment. If so, be sure a SoftRater package has been created and loaded to the proper environment.

Using a Style Sheet Not Created in RateManager

In order for SoftRater to use an existing mapping (i.e. a mapping not created in RateManager), the mapping must be imported into RateManager. Unlike mappings created in RateManager, all custom mappings are saved at the global level regardless of whether they were imported at a program version level or at a global level. If the mapping changes the mapping will need to be re-imported into RateManager.

To Import a Mapping into RateManager

1. Navigate to the Global Inputs Listing where you want to map inputs.
2. On the **Global Inputs Listing** screen, click **Map Custom XML**.
3. This opens the **Input Mapping** screen.
4. Click **Upload Custom Input Mapping**. This opens the **Upload Custom Mapping** box.
5. Enter a unique name for the mapping and then enter the location to the mapping. Click **BROWSE…** to quickly locate the file. When you are finished, click **Save**.
6. The mapping will be imported and can now be used when rating files in the Testing module. It is not necessary to create a new package before using the mapping. See Rating a File with a Mapping for more information.

Chapter 28

PACKAGING

At the end of program development or when a change is made to a program, a process called creating a package must be performed to update the SoftRater rating engine with the logic and tables built in RateManager. A package is a small file that holds all the RateManager logic for a specific program and version.

RateManager supports two kinds of packages:

- **RateManager:** The first is called a RateManager package (*local package)*. A customer hardware environment may include many different SoftRaters. During the RateManager installation, one SoftRater is always designated as "RateManager". If a SoftRater environment exists in the authoring environment and can be rated against using Testing. When a RateManager package is created, it is automatically loaded to the SoftRater environment.

- **SoftRater**: The second type of package is called a SoftRater package (*full package)*. A SoftRater package holds all the same data as a RateManager package, but it also contains extra information that allows the package to be moved between SoftRaters. SoftRater packages are usually created to move a program to a different Rating environment. For example, a package may be moved from Development to QA or from QA to Production.

NOTE: *Creating a SoftRater package also creates a RateManager package.*

CREATING A PACKAGE

From many places in RateManager, you can create a package by clicking **Create Package** on the top navigation bar. The Program ID number is generated and assigned the first time a package is created for a program.

NOTE: *With the exception of changes to data tables for table variables, you must create a new package every time you make a change to a program, prior to rating. If you only make changes to the data tables for table variables, a new package does not need to be created prior to rating. This is* ***only applicable for RateManager Packages not for*** *SoftRater ones.*

The steps below outline creating a package from the program details tab. The same steps (excluding 1 and 2) can be followed to create a package from the following tabs:

- Program Details
- Program Explorer
- Sequencing
- Output Mapping
- Edit Worksheets
- Edit Output Mapping
- Edit Releases

On these tabs, click **Create Package** on the navigation bar.

1. Navigate to the **Program Details** tab for the project, program and folder(s) that contains the program version you want to package.

2. Select the version you want to package and click **Create Package**. A Create Package screen opens allowing you to set options for the package.

   If "Enforce Release Packaging" is set to True in Preferences, the Create Package screen has additional options that can be set.

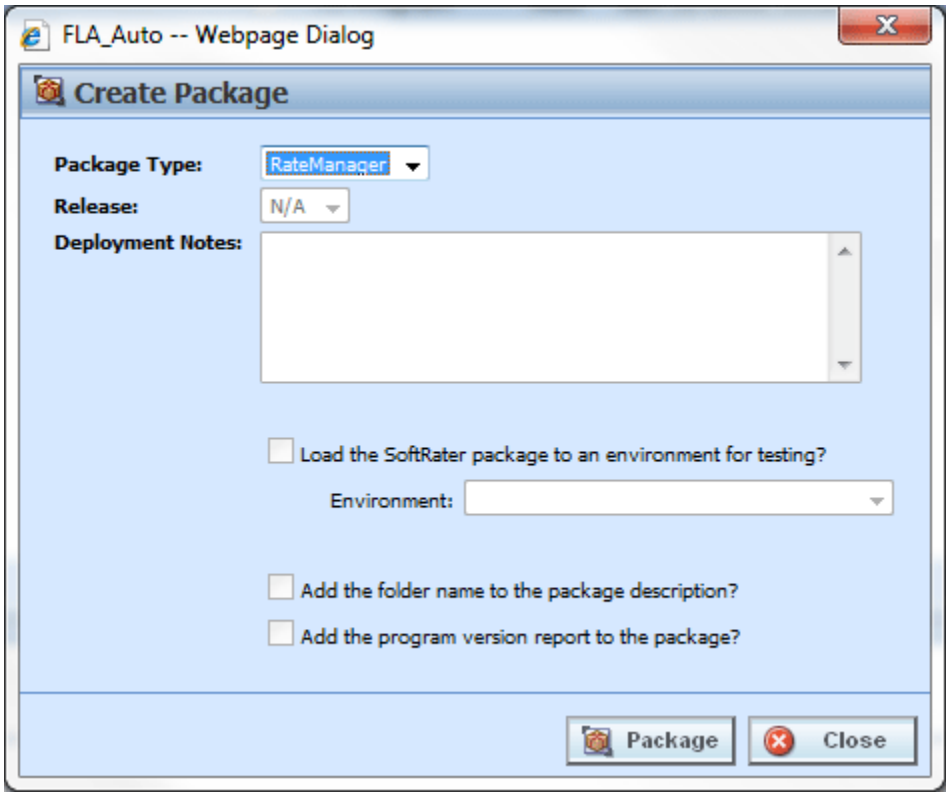


Figure 531 Creating a Package

3. When you have finished setting the package options, click **Package**. The packaging process will start and the progress will be shown. A summary will be shown when packaging is finished.

4. If you created a RateManager package, you can navigate to Testing to rate input files. If you created a SoftRater package, you can navigate to the SoftRater Explorer to load the package. If you encounter errors while packaging, see Resolving Packaging Errors.

NOTE: *You will not be able to create a Packaging that does not have any algorithms in the sequence or results mapped.*

Screen Details – Enforce Release Packaging Set to True:

System administrators set override options. If the system does not allow for overrides, the create global versioning package options screen has a different look. You cannot select the release or the environment to publish. All other options are available.

Package Type: Select the type of package to be created:

- RateManager
- SoftRater

Release: If you are creating a SoftRater package, select the release where the package will be placed. After the release is deployed, the package will be available for download by any environment on the network that has the security authorization to see the packages that have been created.

NOTE: *If the Enforce Release Packaging is set to True, users have the option to select the Release and not the Environment when they go to package.*

Deployment Notes: If you a creating a SoftRater package, you can enter comments about the package. The comments are viewable from SoftRater Explorer in IBFA.

Load SoftRater package to an environment for testing: Check for Yes, leave blank for No.

Environment: If you checked Yes for loading SoftRater package to an environment for testing, you must select the environment where you want to load the package for testing.

Add the folder name to the package description? If you are creating a SoftRater package, check this box to add the folder name to the SoftRater package description. This feature allows IBFA Administrators to differentiate packages.

Add the program version report to the package? Adds a program version report to the package. A program version report provides the details of a program version.

Add input/output information: only displayed when input and output mapping files are found on the server.

Add the input mappings to the package?: Add the input mappings to the package. This option is only displayed when input and output mapping files are found on the server.

Add the output mappings to the package?: Add the output mappings to the package. This option is only displayed when input and output mapping files are found on the server.

Screen Details – Enforce Release Packaging Set to False:

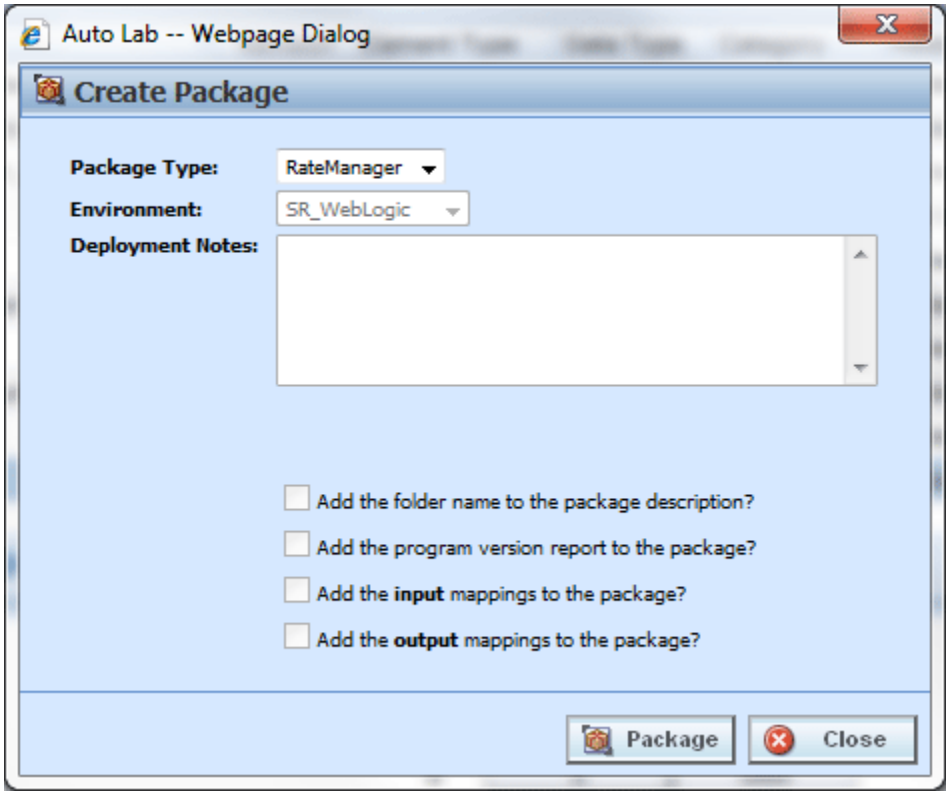


Figure 532 Creating a Package Release Packaging Set to False

Package Type: Select the type of package to be created:

- **RateManager**
- **SoftRater**

Environment: If you are creating a SoftRater package, select the environment where the package will initially be placed. After packaging, the package will be available for download by any environment on the network that has the security authorization to see the packages that have been created.

Deployment Notes: If you a creating a SoftRater package, you can enter comments about the package. The comments are viewable from SoftRater Explorer in IBFA.

Add the folder name to the package description? Check if you would like to add the folder name to the SoftRater package description.

Add the program version report to the package? Check if you would like to add the program version report to the SoftRater package. See Program Version Report.

Add the input mappings to the package?: Add the input mappings to the package. This option is only displayed when input and output mapping files are found on the server.

Add the output mappings to the package?: Add the output mappings to the package. This option is only displayed when input and output mapping files are found on the server.

VERSIONING PACKAGE

In order for SoftRater to automatically choose the version of a program to rate, a versioning package must be created. A versioning package is universal to all programs under a particular product. Only one versioning package needs to be created for each product. Create a new versioning package anytime you make changes to the way program versions are selected for a program.

Versioning packages are created the same way program packages are created, except they are created from the project. For more information on versioning, see Versioning.

Global Versioning logic uses the program versioning criteria to determine which program version should be used by the program being called.

Creating a Versioning Package: Utilizing Releases

System administrators set the option to force releases to be used for package deployment. If you want to select a different environment for loading a SoftRater package to, the override option must be set to true.

To create a versioning package, right click any product and select Create Versioning Package.

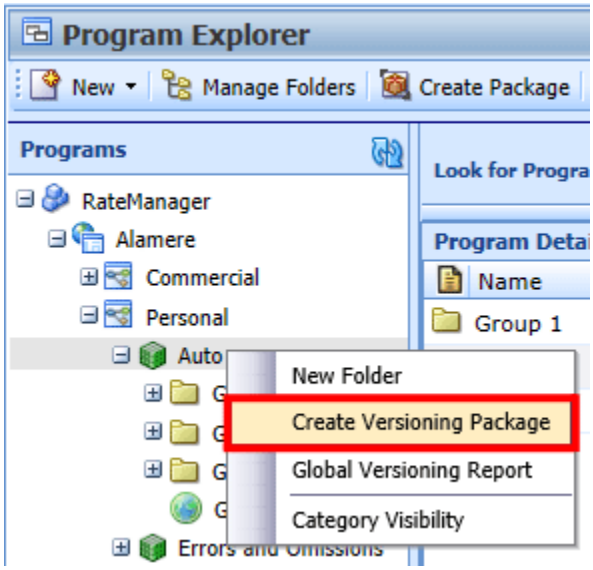


Figure 533 Creating a Versioning Package

This pulls up the global versioning package screen and where you can create a versioning package for that product. This feature can be used at any time and does not require you to leave the screen you are currently working in.

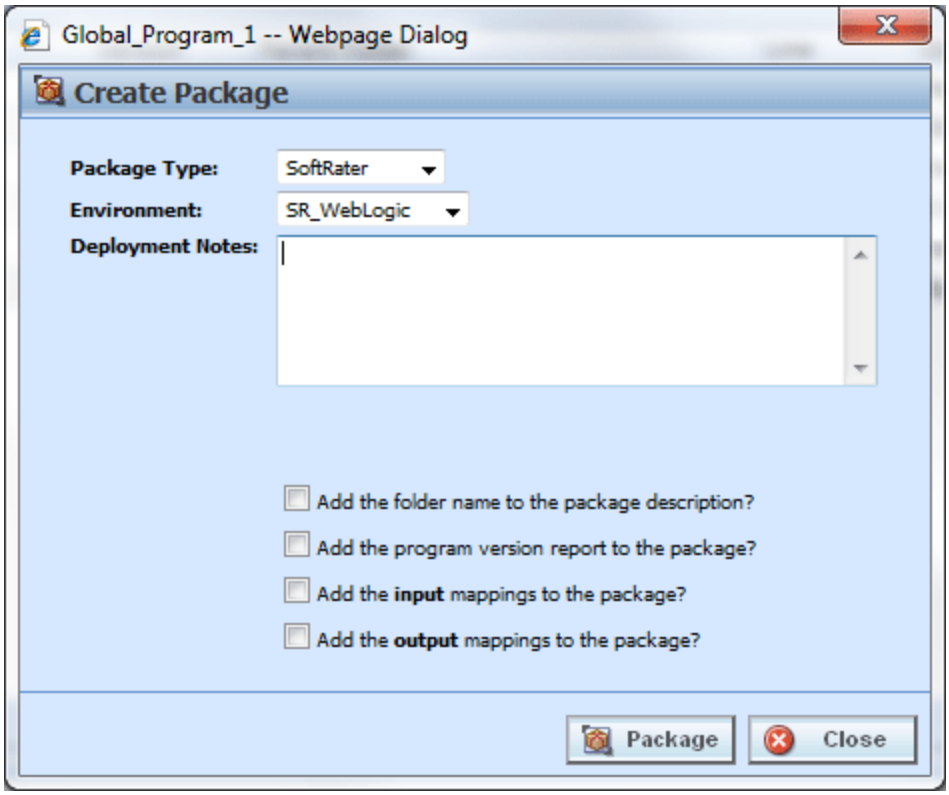


Figure 534

3. When you have finished setting the package options, click **Create Package**. The packaging process will start and the progress will be shown. A summary will be shown when packaging is finished.

   Package Type: Select the type of package to be created:

   - **RateManager**
   - **SoftRater**

   Release: If you are creating a SoftRater package, select the release where the package is initially placed. After the release has been deployed, the package will be available for download by any environment on the network that has the security authorization to see the packages that have been created.

NOTE: *If the Enforce Release Packaging is set to True, users will have the option to select the Release and not the Environment when they go to package.*

Deployment Notes: If you a creating a SoftRater package, you can enter comments about the package. The comments are viewable from SoftRater Explorer in IBFA.

Load the SoftRater package to an environment for testing: Check for Yes, leave blank for No. This option will be available if the system has been setup to allow release override.

Environment: If you checked Yes for loading SoftRater package to an environment for testing, you must select the environment that the testing package will be loaded to. This option is available if the system has been setup to allow release override.

Add the folder name to package description? Check if you would like to add the folder name to the SoftRater package description.

Add the program version report to the package? Check if you would like to add the program version report to the SoftRater package. See Program Version Report.

Add the input mappings to the package?: Add the input mappings to the package. This option is only displayed when input and output mapping files are found on the server.

Add the output mappings to the package?: Add the output mappings to the package. This option is only displayed when input and output mapping files are found on the server.

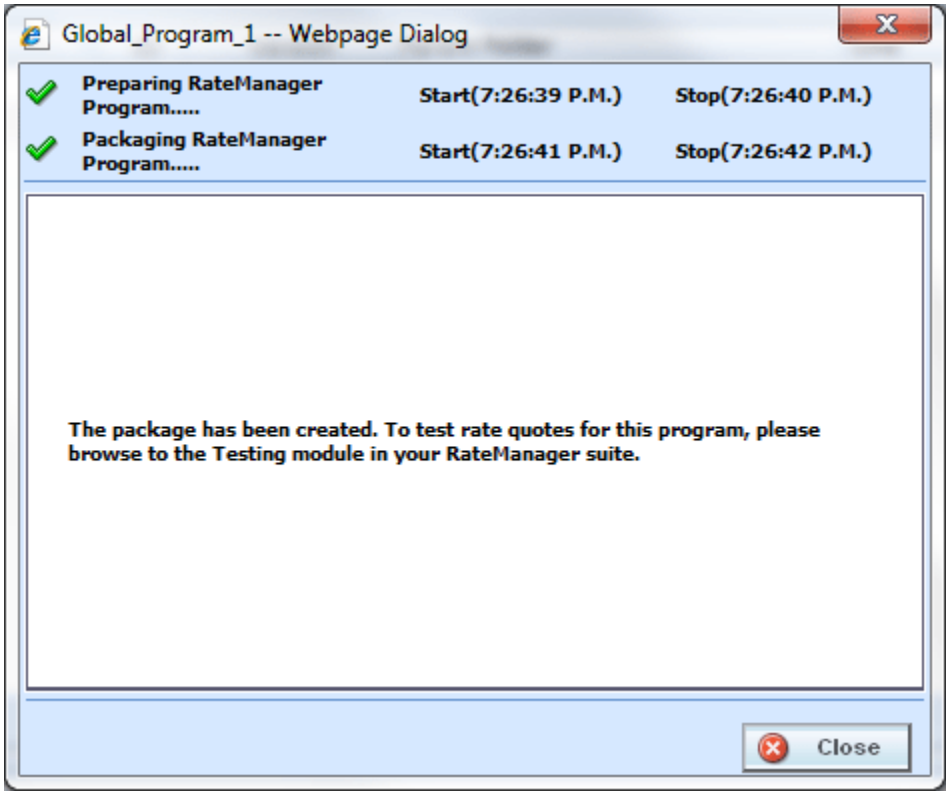


Figure 535 Global Packaging Success Message

4. If you created a RateManager package, you can navigate to Testing to rate input files and test version selection. If you created a SoftRater package, you can navigate to the SoftRater Explorer to load the package.

Creating a Versioning Package: Not Allowing for Override

System administrators set override options. If the system does not allow for overrides, the create versioning package options screen will have a different look. You will not be able to select the release or the environment to publish. All other options will be available.

Package Type: Select the type of package to be created:

- **RateManager**
- **SoftRater**

Environment: If you are creating a SoftRater package, you can select the environment where the package is initially be placed. After packaging, the package will be available for download by any environment on the network that has the security authorization to see the packages that have been created.

Deployment Notes: If you a creating a SoftRater package, you can enter comments about the package. The comments are viewable from SoftRater Explorer in IBFA.

Add the folder name to the package description? Check if you would like to add the folder name to the SoftRater package description.

Add the program version report to the package? Check if you would like to add the program version report to the SoftRater package. See Program Version Report.

Add the input mappings to the package?: Add the input mappings to the package. This option is only displayed when input and output mapping files are found on the server.

Add the output mappings to the package?: Add the output mappings to the package. This option is only displayed when input and output mapping files are found on the server.

RESOLVING PACKAGING ERRORS

There are many reasons why RateManager won't allow a program to be packaged. Usually, however, the reason is simple and resolution is easy. Listed below are the most common packaging errors.

1. **There are no results mapped.** RateManager requires that your program have at least one result (output). This error means you don't have any result groups defined or that there are no results in the result group that is defined. To fix this problem, create results for the program and try the package request again. See Introduction to Result Mapping for more information.

2. **There are no algorithms in the sequence.** RateManager uses the sequencing to determine which elements (algorithms and variables) to process and the order in which to process them. This error means that you haven't added any algorithms to the program sequencing, or that none of the algorithms in the sequencing are activated. To resolve this error, add at least one algorithm to the program sequencing or activate one of the algorithms already there. See Sequencing for more information.

3. **At least one of the algorithms in the sequencing does not have an input used for the category.** SoftRater uses inputs to determine which categories need to be processed. If you receive the error below, it means that you have an algorithm in the sequence whose category

does not have any inputs used.

If there are no inputs associated with a category, this means that there is a category level elements in the program sequencing but that none of the elements in the sequencing uses that category level input.

To resolve the issue, you can do one of two things:

1. Deactivate all algorithms in that category.
2. Use an input for the category. (An algorithm does not have to directly use an input. The input can be used in a variable that is used in another variable, etc. that is eventually used by an activated algorithm.) No inputs are required for the Driver-Vehicle category.

If a package fails to complete, an error message will be displayed.

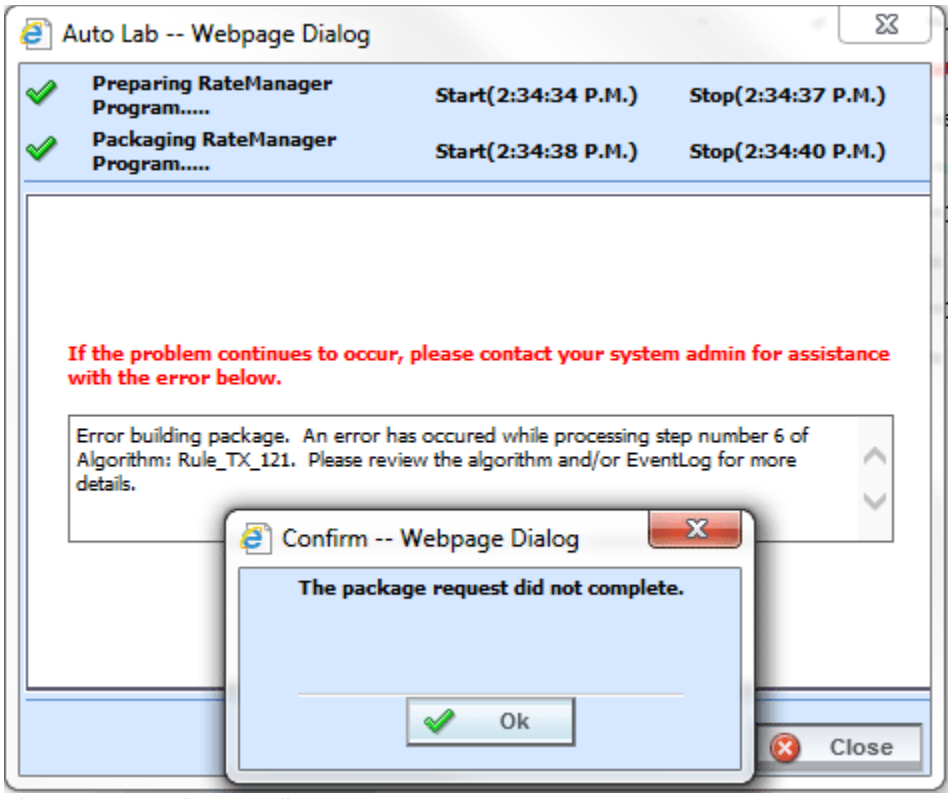


Figure 536 Package Failure

TEST PROGRAM

The Test Program tab gives you the ability to create, execute, debug and maintain test scenarios within the program version where you are currently working. Testing and debugging of a program can either be done within the program using the Test Program tab, or using the Testing Module.

Test files are created and maintained within the Test Program tab using either the XML view or the grid view with text boxes for input values. Additionally, new and existing test files created outside of RateManager are accessible using the Test Program tab, or you can paste and rate your own XML files. While it is possible to test multiple programs, you will not be able to save any changes to programs outside of the program where you are currently working. It is recommended that you test programs outside of the program version where you are currently working in the Testing module.

Programs must be packaged before going through Test Program. If you enter Test Program and the only category is Policy, then you may need to package the program. When you try to rate a program that has not been packaged, you will receive an error. If there is no package date, package your program and try again.

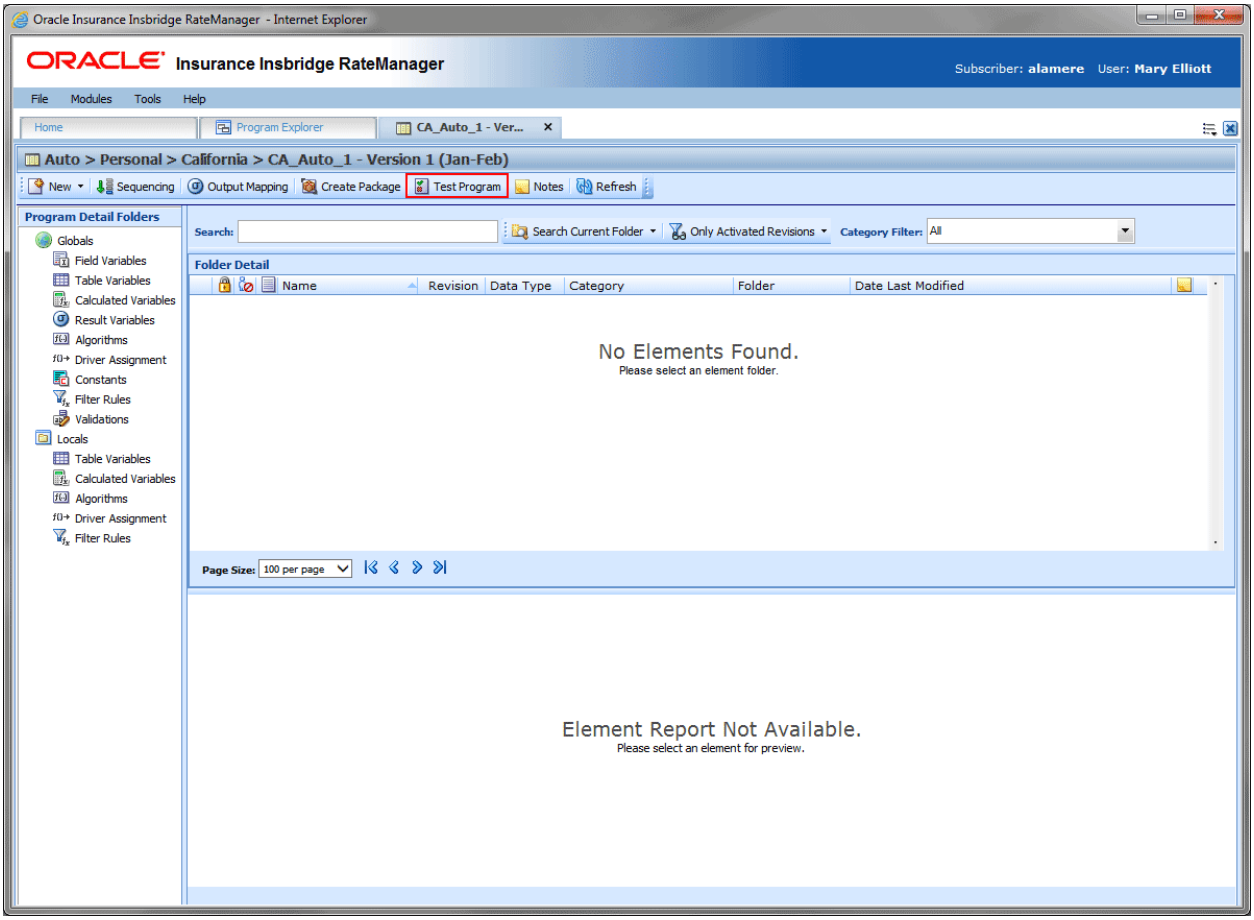


Figure 537 Selecting Test Program

Navigating to Test Program Tab

Test Program is not an option on the Program Explorer tab.

1. Modules→Program Explorer. This opens the Program Explorer tab.
2. Select the domain type→domain→project→Folder→Program→Program Version.
3. **Open** the program version to get the Program Details tab. Test Program is available from the menu bar at the top of the tab.

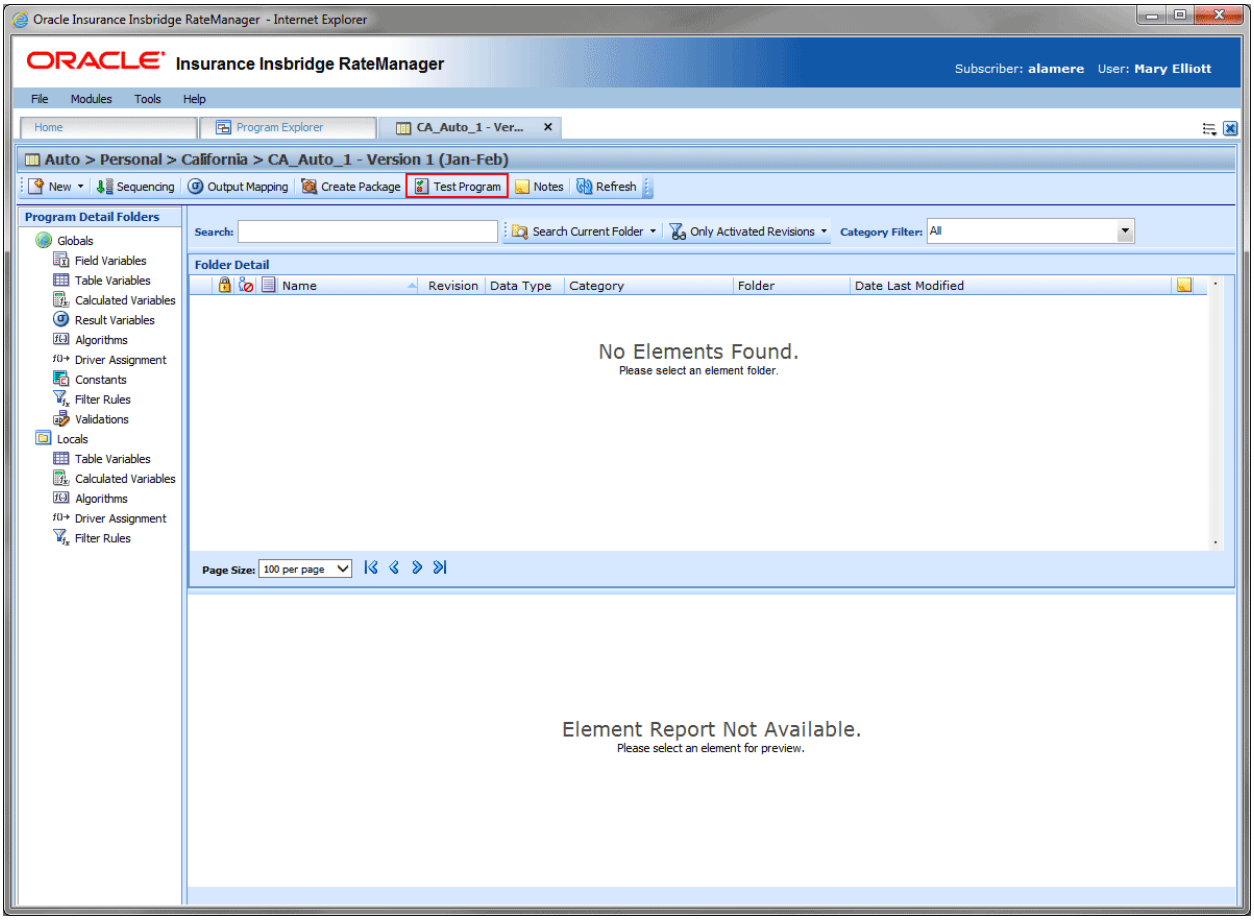


Figure 538 Test Program Tab

Test Program may also be accessed through Program Sequencing and Output Mapping tabs. Be aware that if you have made any changes to sequencing or mapping, you will need to package the program prior to using Test Program.

There are two sub-tabs on the Test Program tab:

- **Test Case**: Displays the program inputs and where you can enter values for testing.
- **Debug Report:** A detailed report of the rating.

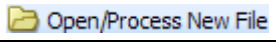
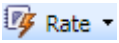
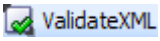
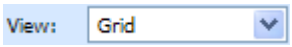
TEST CASE TAB

The Test Program tab has two sub-tabs: Test Case and Debug Report. By default, the Test Program tab opens on the Test Case sub-tab.

FUNCTIONALITY

The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform on this tab.

Functions:

	New: This option will clear all fields and allow you to create a new test case. You will be asked if you want to save any previous entries. See Creating a Test Case for more information.
	**Open:** Browse for existing test cases.
	**Rate:** Rates the current test case. Filter rules and validation rules are automatically included in debug results. This option can be unchecked if those results are not needed. The default is for this option to be on.
	**Validate XML:** Available in XML View only. Validates your XML entries to make sure they are formatted correctly. Values are not validated only format.
	Save: Saves the current test case. This option is only for test cases within the program version.
	**View:** Switches between XML view and Grid view. When opening a new test case, Grid view is always the default.

Screen Details

Category Selection: In Grid View, the left side of the tab will contain the categories of the program. You also can add, delete and copy categories as needed. In XML View, this area will not be displayed.

In grid view the RATE, HEADING and POLICY nodes will be present.

- The RATE node contains the product and the rating environment information. The default environment will be used unless you select another environment.
- The HEADING node contains the program information, parent ID, program ID and version. You will not be able to change this information.
- The POLICY node contains the inputs for the policy category. You can make as many changes as you need. Other categories will be listed underneath.

Input Assignment: In Grid View, after making a selection from the categories selection area, the input assignment area will be populated with the inputs from the program. You can enter values directly into these fields. In XML View, the tab will display an XML file where you can enter values.

Views

There are two views available. Grid view is the default.

Grid View

In Grid view, the left window of the Test Program will show the categories and category structure created for the program. When a specific category is clicked, the right side of the tab shows the inputs available for the chosen category.

See Creating a Test Case in Grid View for more information on creating a test case in Grid view.

XML View

In XML view, an editable XML file is displayed. You can edit this file by typing values into the “V” attributes of each available input. You also will be able to cut and paste.

See Creating a Test Case in XML View for more information on creating a test case in XML view.

Right Click Options

In Grid View, the Category Selection area on the left side has a right click menu where you can add, copy, or delete. The Heading and the Policy category do not have a right click menu.

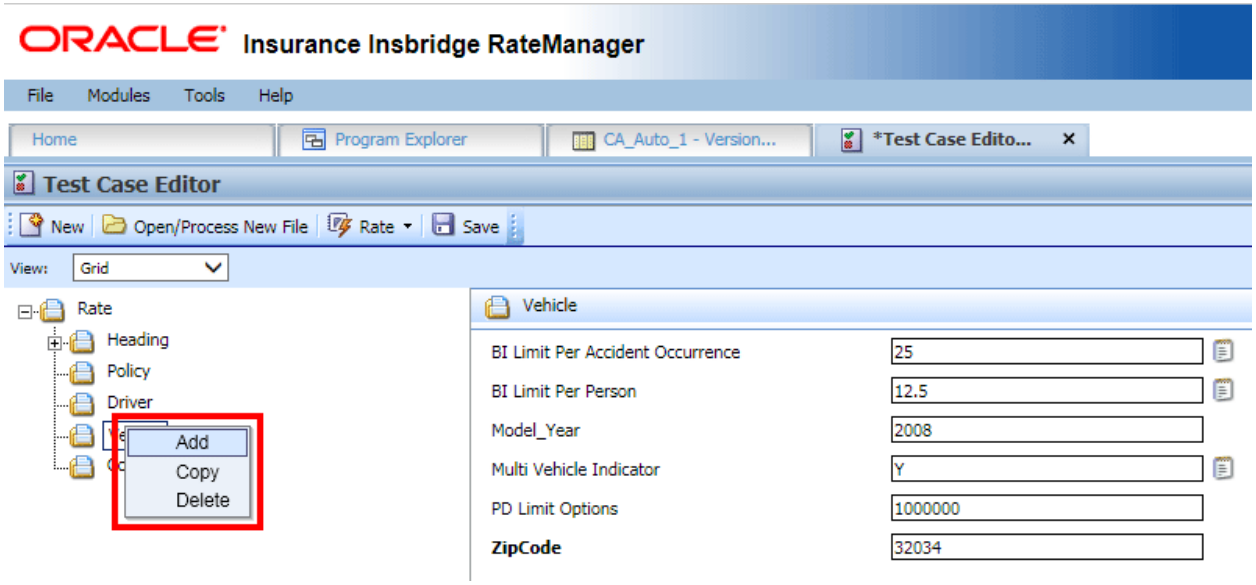


Figure 539 Right Click Menu for Test Case

- **Add**: Adds another category. Use this when you have to account for multiple entries in the same category. For example, you may need to add another vehicle category to accommodate two cars in the rating.
- **Copy**: Use this when you want to copy a category and the contents. For example, if you have two vehicles and with only two differences between them. You can complete the category and then

use the copy feature to copy the category. On the second entry you can change the differences.

- **Delete**: Use this to remove a category that is no longer needed.

The XML View has a right click menu that allows for select all, cut, copy, paste, delete and undo.

Creating a Test Case in Grid View

1. Navigate to the Test Program tab for the desired program.

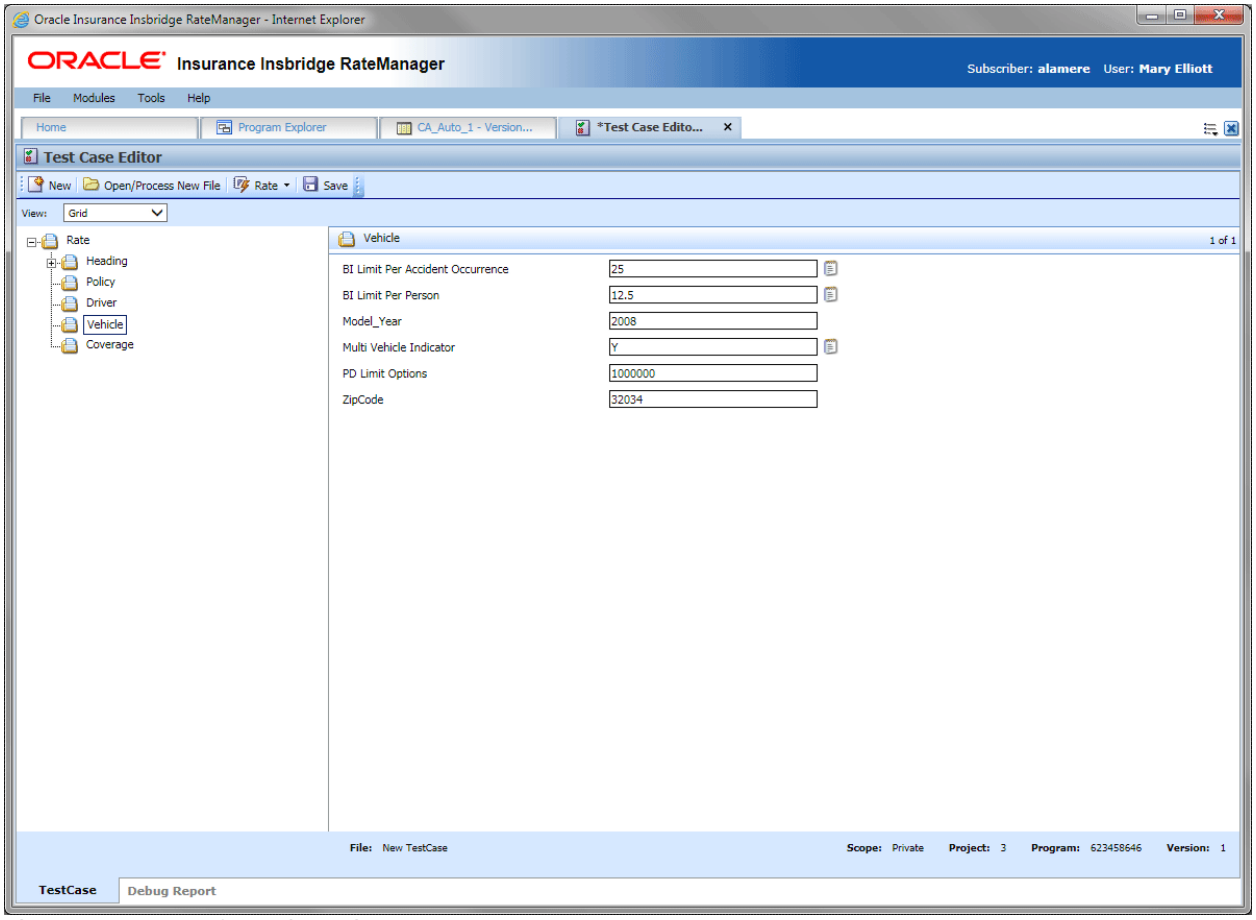


Figure 540 Inputs for Selected Category

2. If not already displayed, choose Grid from the view menu.

3. In the left window, click the category name where you want to enter data. The right window will display the applicable inputs for the chosen category.

4. In the right window, you will have two options, entering a value in directly or selecting from a pre-populated list. You also will be able to copy and paste entries.

NOTE: *Hold your cursor over the input to view the input data type.*

   a. To enter in a value directly, select the field and type in your entry.
   b. To select a value for string input types, click the icon at the end of the field. This will bring up a menu.
   c. The date format used by Test Case Editor uses the format YYYY-MM-DD. This may not reflect the date format used by the program.

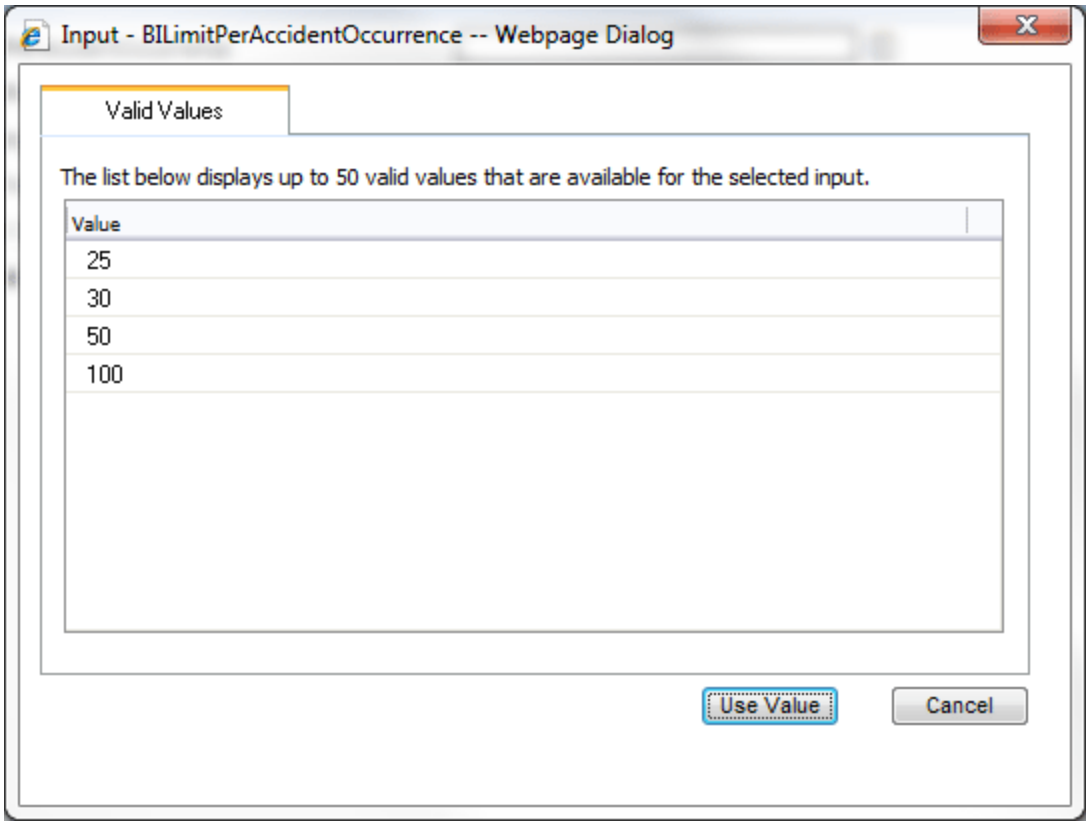


Figure 541 Entering Inputs in Test Program

d. Select the option you want. Click **USE VALUE**. This closes the window and returns you to test case. Your selection will be filled in the field. If no valid values have been entered for this field, the list will be empty. You will have to enter a value in directly.

NOTE: *You also can enter in a value for string inputs. You are not required to use the notepad icon.*

5. Repeat Steps 3 and 4 for each category and input for the test case.
6. When you have finished entering your inputs, click **RATE** to rate the test case.
7. The Test Case sub-tab switches to the Debug Report sub-tab once the test case has finished rating. See Using the Debug Tab for more information.
8. If you would like to save your test case, return to the Test Case sub-tab.
9. Click **Save** to save the test case.

Retesting

You can clear the fields and begin your test case again by clicking **New** in the top menu bar. This clears the fields and allows you to enter new values for testing. Any new or deleted categories will also be cleared or added back to the test case.

Prior to clearing the fields, you will be asked if you want to save your current entries.

Saving a Test Case

Only test cases that are within the program version can be saved. Non program version test cases cannot be saved in the Test Program tab.

The save screen is where you can specify a unique name for the test case. Unique names allow you create multiple scenarios. For example, you can use one set of values for your inputs and save that test case. Then change the values and save the test case under another name. You can save as many test cases as you need. Saved test cases will be listed in the body of the Server Files window. All test cases saved in Test Program will be saved to the server.

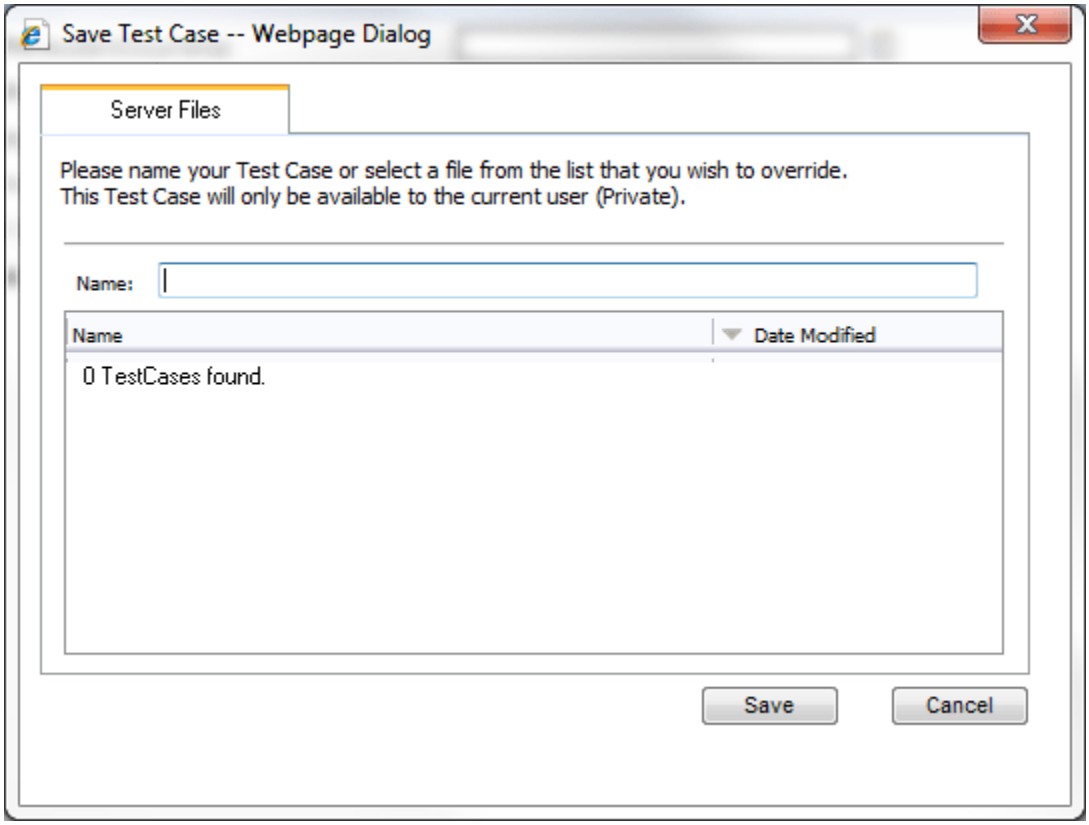


Figure 542 Saving Test Case Files

If you would like to work with your test case on the Testing module, make sure you save your file. Files that are saved to the server also will be available on the Testing module.

Creating a Test Case in XML View

In XML view, an editable XML file is displayed. You can edit this file by typing values into the “V” attributes of each available input. You also can delete or add inputs and you will be able to copy and paste. Inputs can be validated for proper formatting. Input values are not validated.

XML view is helpful if there is a child category where the parent category does not have an entry. All categories are visible in XML view.

1. Navigate to the Test Program tab for the desired program.

2. Choose XML from the view menu.

3. An editable XML file of the program will be displayed. You may manually edit this XML file to create a test case.

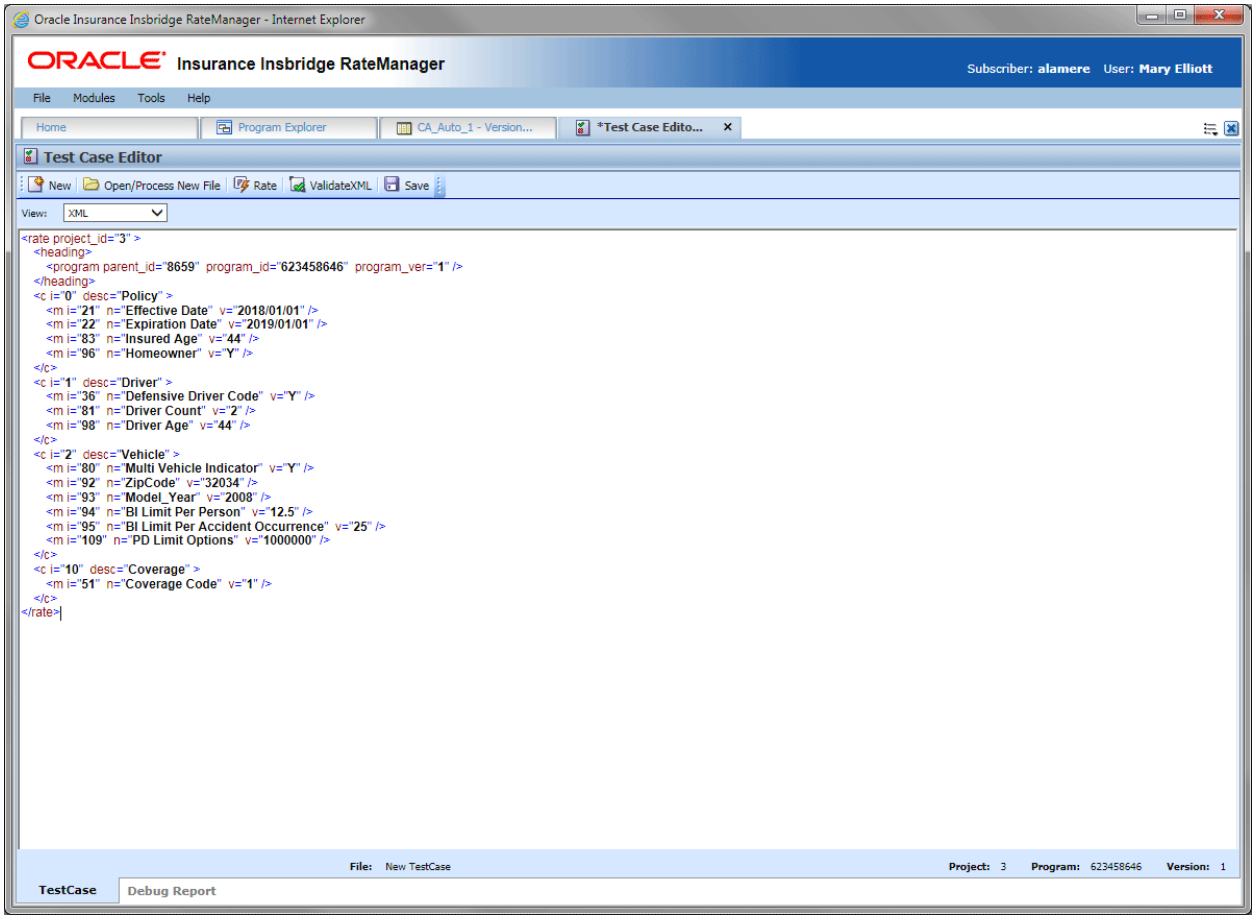


Figure 543 Test Case XML View

4. Click **Validate XML** to validate the XML before rating the test case.

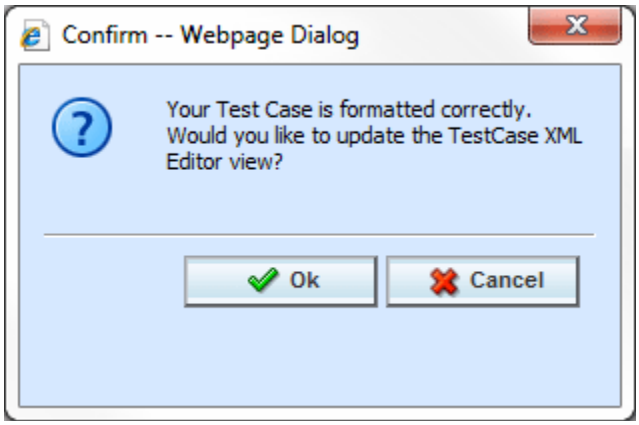


Figure 544 XML Validation

5. If the file is formatted properly, you will receive an update message. If there is a problem with your file, you will be instructed to make changes. Click OK when your file is ready.

6. Click **Rate** to rate the test case.

7. The Test Case sub-tab switches to the Debug Report sub-tab once the test case has finished rating. See Debug Tab – Toggling Views for more information.

8. If you would like to save your test case, return to the Test Case sub-tab

9. Click **Save** to save the test case.

DEBUG TAB

You will have the ability to view debug reports from the Test Program tab without having to go to the Testing Module. Rating a test case switches tabs to the Debug Report sub-tab.

There are two ways the debug report can be presented:

- **Single Format:** A single report format is where the entire report is listed out. This is the default style in RateManager. All algorithms with execution details are listed in execution order. A single format may be useful when smaller programs are run or when all execution details need to be listed on a single page.

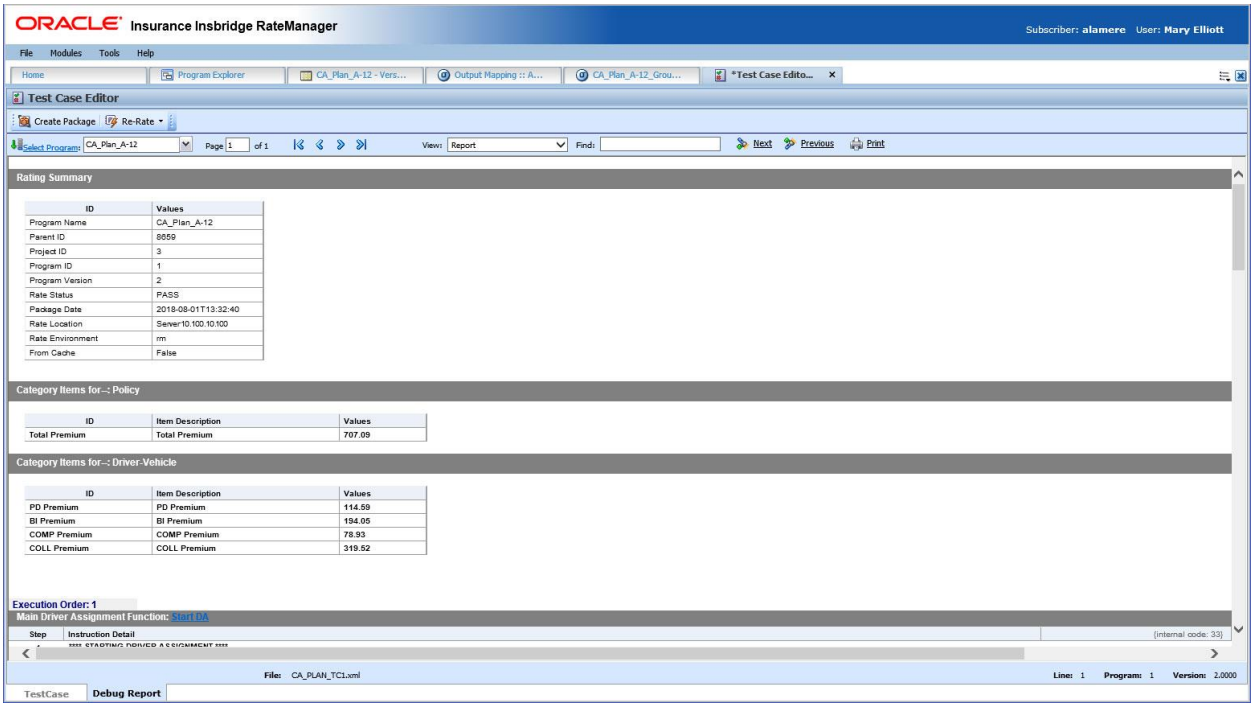


Figure 545 Debug Single Report Format

Rating Summary information will be at the top, followed by detailed Category Information. The execution order is listed next.

- **Summary Format:** A summary report format is where information is contained in sections that can be displayed or hidden. The summary format also uses pop out windows. The pop out window contains all the debug information for the selected algorithm. Multiple pop out windows can be open simultaneously. Summary report format may be useful if large programs are to be run.

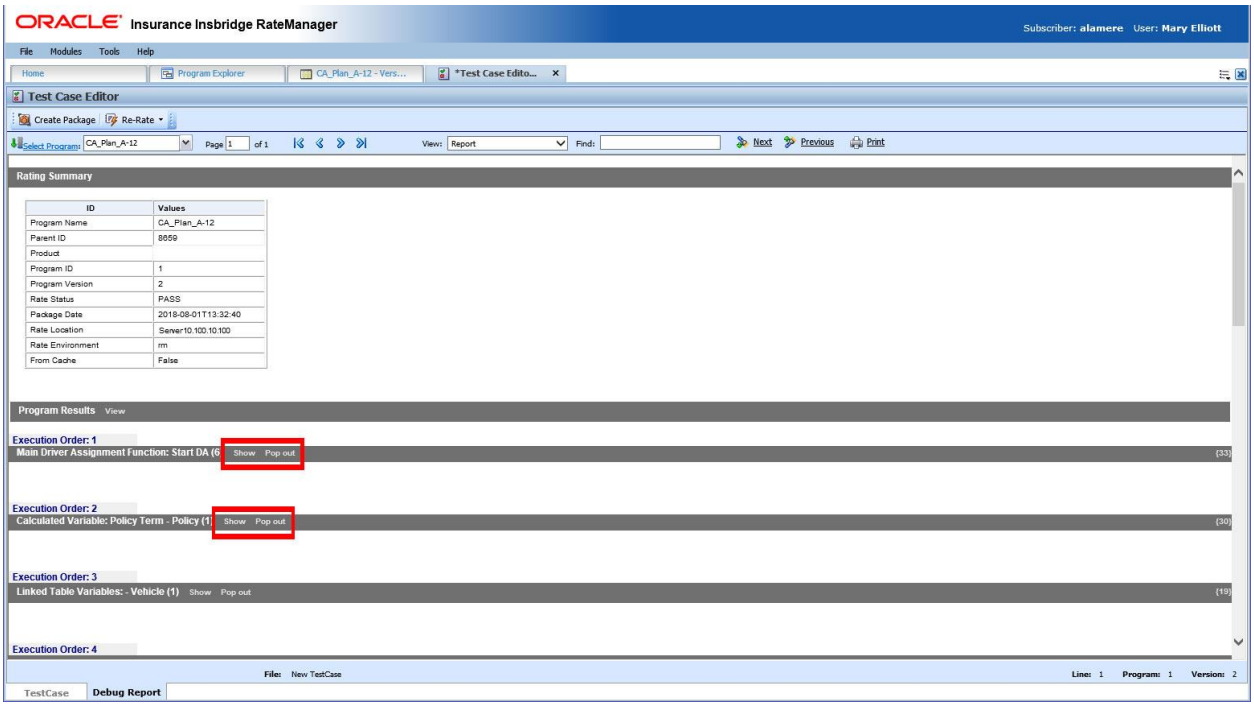


Figure 546 Debug Report in Summary Format

Rating Summary information is at the top, followed by detailed Category Information. The execution order is listed next.

Details for each section can be shown in the report and then hidden or displayed in a pop out window.

Figure 547 Debug Report Pop Out Window

The setting for the report style is in the Tools→Preferences window and can be updated at any time. The Preferences window requires admin privileges. The setting applies to the entire system and all users. If you need to update the report format and do not have access to the Preferences windows, please contact your system administrator.

Toggling Views

You can toggle between the test case entries and the debug report by clicking to tabs. The information on each tab is held until you close Test Case Editor.

The current tab is bold, the other tab is grayed out.

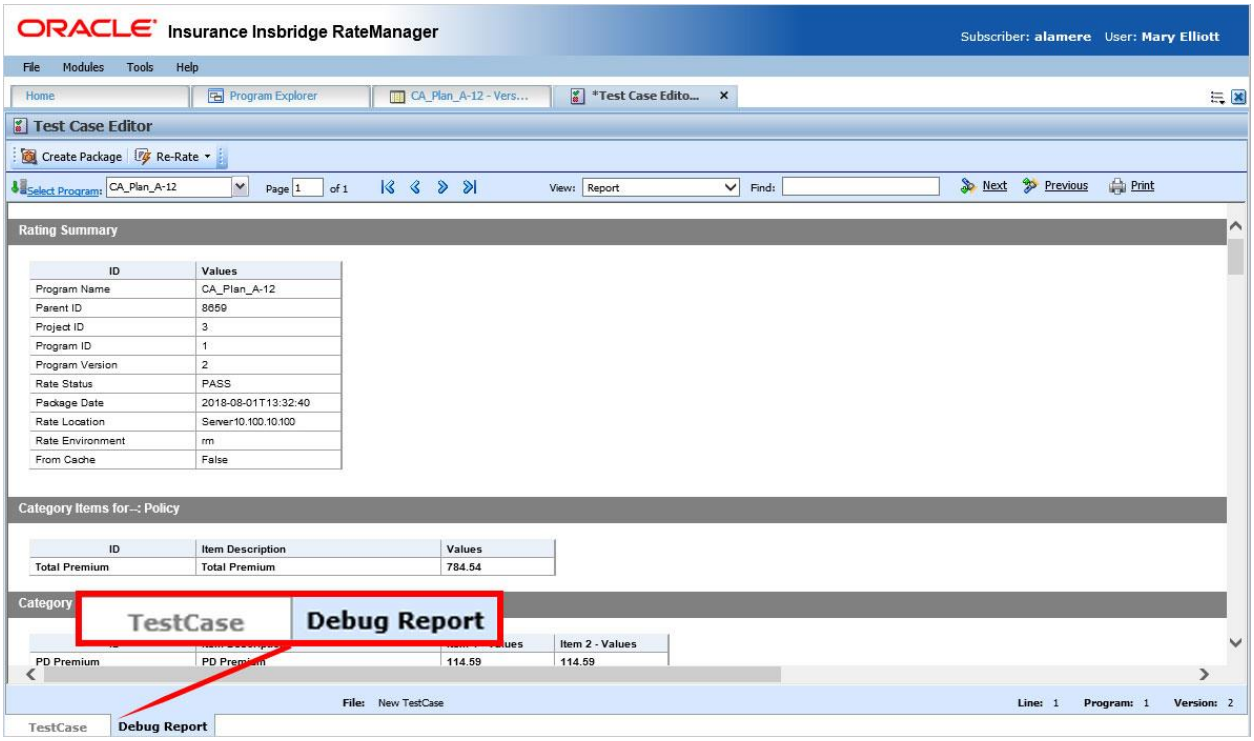


Figure 548 View Debug Reports

Functions

Create Package	**Create Package:** Creates a package of the selected program version.
Re-Rate	**Re-rate:** Re-rate rates the selected test case and replace any preexisting debug report with the newly rated one. Filter rules and validation rules are automatically included in debug results. This option can be unchecked if those results are not needed. The default is for this option to be on.
Select Program:	**Select Program:** Opens a separate screen where programs, listed in a tree view, can be opened in a new window.
Program Name: 75202	**Program Name:** Displays the name of the program being rated. The drop down menu is populated when you are rating multiple programs.
View: XML	**View:** The debug tab automatically shows results in a report view. You can change this to show the raw XML output and if available, worksheet XML or sample worksheet format.
Find:	**Find:** Search through the debug report for a specific alpha/numeric character string.
Next	**Find Next:** If a search is performed, Find Next will navigate to the next place in the debug report that a match was found.
Previous	**Find Previous:** If a search is performed, Find Previous will navigate to the previous place in the debug report that a match was found.
Print	**Print:** Prints the debug report to your local or network printer.

Using the Select Program Link

The Select Program link can help to navigate P2P callout programs. Clicking the Select Program link opens a separate window where called programs are listed in a tree view. Programs are appended with a sequenced number to identify the programs in the order they are called.

Select the program you want to view and click Open to open the selected program in the same window, or Open in a new window, to open the selected program in a new browser window. Opening in a new window does not close the program tree view. Multiple windows can be open at one time. Opened programs have a locator link or breadcrumb trail in the top dialog window frame to identify the program call sequence.

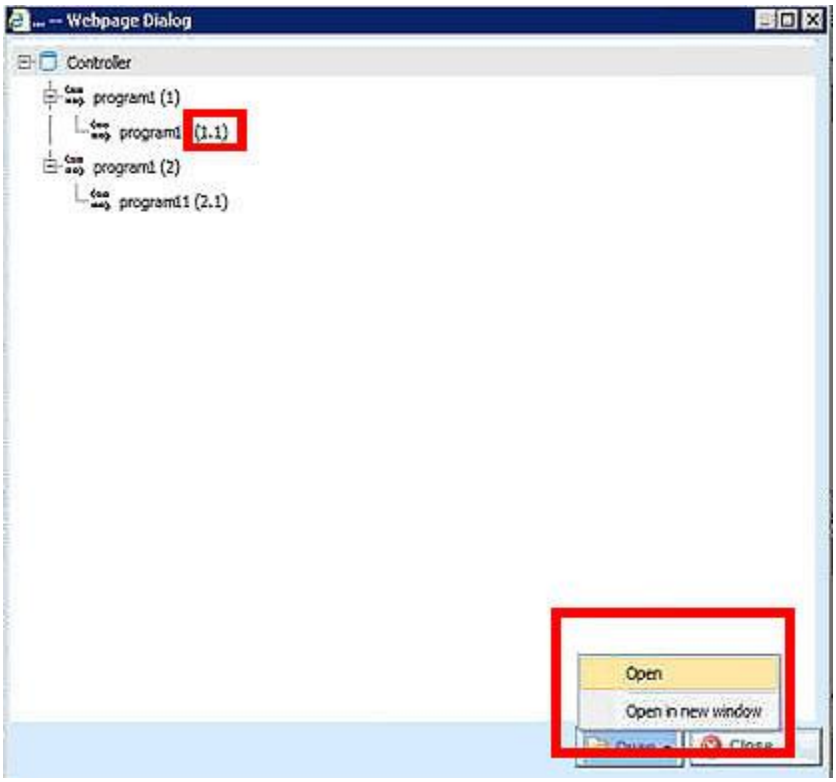


Figure 549 Select Program Popup

Using Find

The Find option searches the content displayed on the screen. When using summary report format, execution details are hidden. Find does not search hidden content. To search a summary debug report, you may need to show the content you want to search.

To use Find, enter a value into the field and click Next. The next instance of the value will be highlighted. If no match is found, a message is displayed.

Screen Details

The screen details vary depending upon the Debug option selected.

If Debug has been selected:

- The Rating Summary is displayed at the top of the screen.
- The Category Items are listed next.
- The Debug Rating Report is last. This area contains the algorithms, drivers assignments, calculated and table variables used listed in execution order with the details of the actions performed and the results.

If Debug has not been selected:

- The Rating Summary is displayed at the top of the screen.
- The Category Items are listed next.

No algorithms will be listed.

In summary report, the screen details are the same but displayed in summary style.

NOTE: *For efficiency purposes, no program to program callout will include empty results.*

Hyperlinks to RateManager Elements

The debug report contains hyperlinks that open an element for editing.

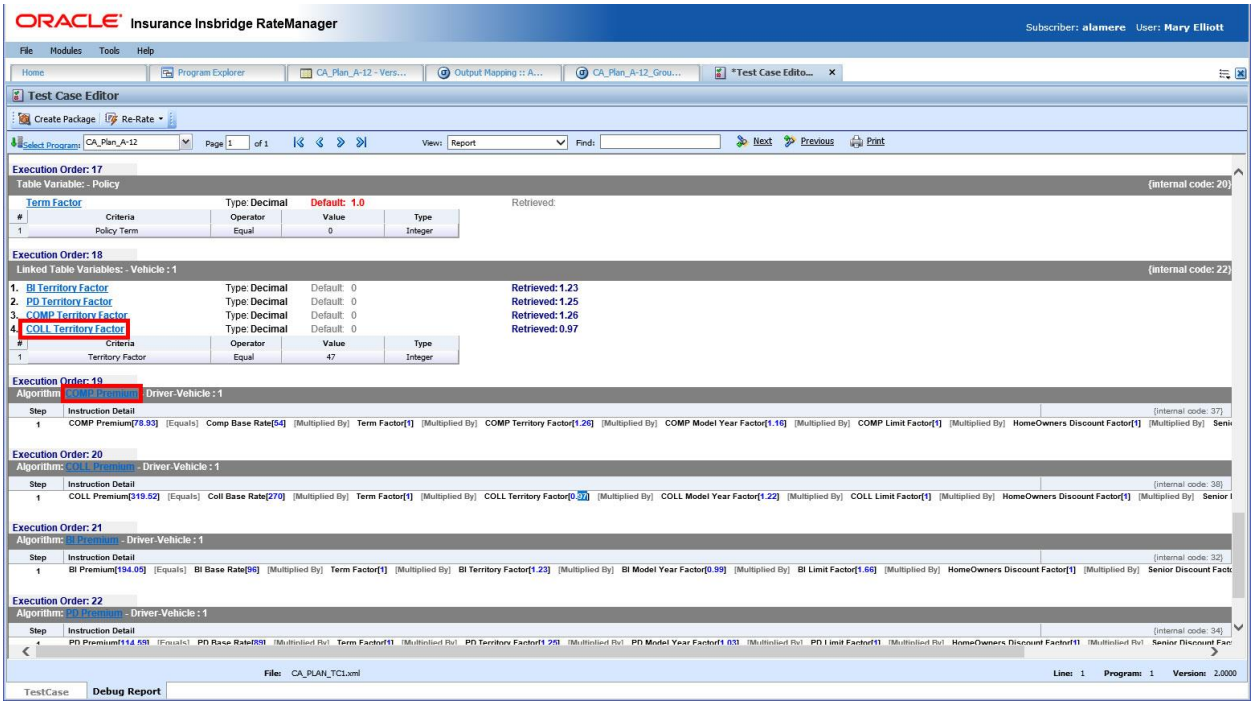


Figure 550 Hyperlinks in Debug Report

Available elements include: algorithms, calculated variables, driver assignments, filter rules, and table variables. Links are located in the name of the element. Clicking the link opens the edit screen for the element. After editing, the program can be re-packaged on the Debug Report tab.

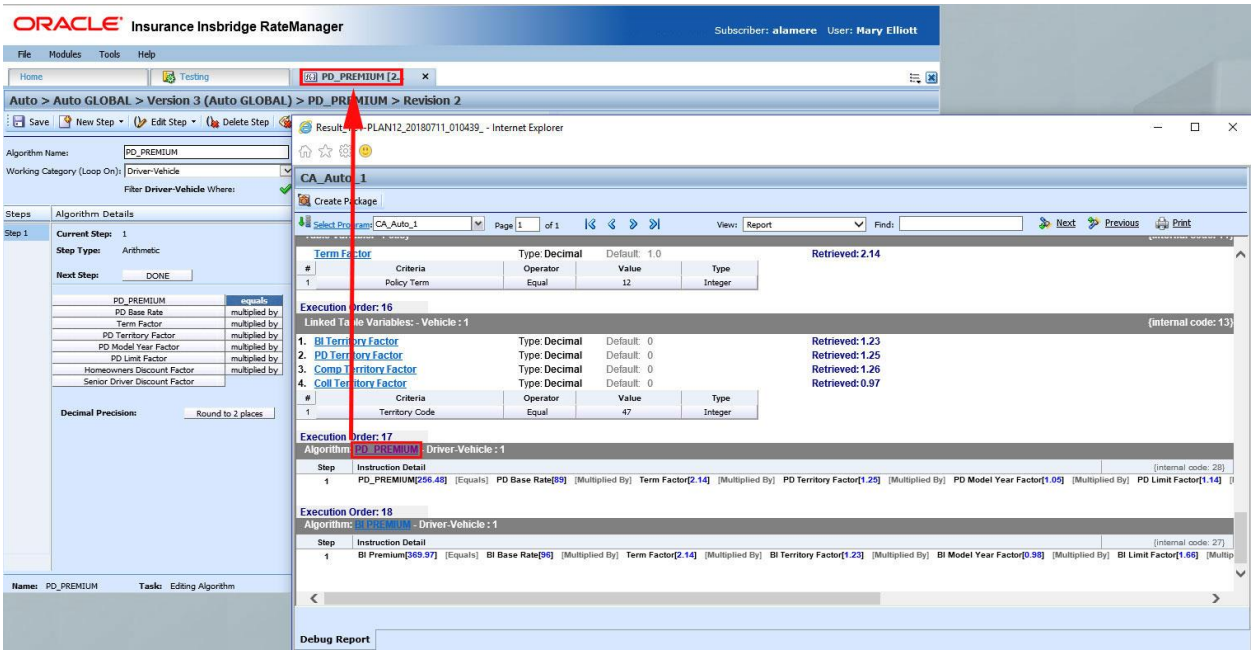


Figure 551 Editing an Element from Debug Report

Write permissions must be in place to edit elements. An error message is displayed if permissions are not in place. Package permissions must be in place to package a program.

Hyperlinks are on the debug report only. No links are available when the report is in XML view, worksheet view or sample worksheet view.

Advanced Options

Advanced Algorithm Looping is displayed on the Debug Report.

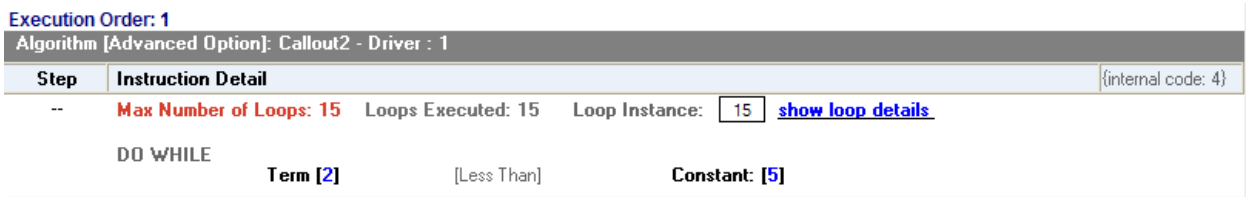


Figure 552 Advanced Options in Debug

In the gray information bar you will see the [Advanced Option] indicator. The DO WHILE condition that caused the looping will be listed, as well as the details of the condition. The maximum number of loops and the number of loops performed will be displayed.

The Loop Instance will show the number of the last loop run. To view the details of this loop, click the hyperlinked **show loop details** link. This will place you on the looping details screen. This screen will only display the details of that loop. You can view any loop run by entering in the number in the Loop Instance box. For example, if you wanted to see the details of loop 9, you would enter 9 in the Loop Instance box

and click the **show loop details** link. The properties of the 9th loop would be displayed in a separate screen.

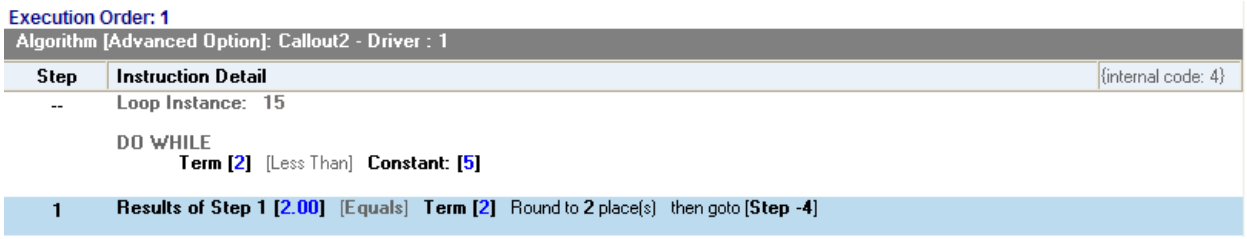


Figure 553 Looping Details

The looping details screen contains the summary policy information and the looping algorithm information. If any result variables were set during the looping, a table will be displayed at the top of the tab, organized by loop iteration.

Create Category Instances on the Debug Report

The Create Category Instance step uses the same basic debug definition as callouts. Each item from the function mapping is available on the View Mapping link.

A DYN_ prefix is affixed to each dynamically created category instance. For example, DYN_2 – (INSTANCE NAME) indicates that this is dynamic instance number 2.

Create Category Instances in Callouts

In P2P and P2L callouts, if a program returns more categories than were sent, the system uses the exception mapping to set the result variables of the new category that was created.

If the callout does not have an exception mapping, the system matches the result variables for the created category.

A system created attribute on the category "c" node, "di", signifies that this category was created dynamically. The value is equal to 1 when true.

```
<result>
  <program>
    <c i="0" d="Policy" di="1">
      <m i="POLICYNUMBER" d="PolicyNumber" v="TX100201"/>
    </c>
    <c i="4" d="Coverage">
      <m i="COVERAGE_SURCHARGE" d="Coverage_Sur" v="3"/>
    </c>
  </program>
</result>
```

Debug Report XML

Debug reports also can be displayed as XML. You can expand and reduce nodes as well as enter values and re-rate.

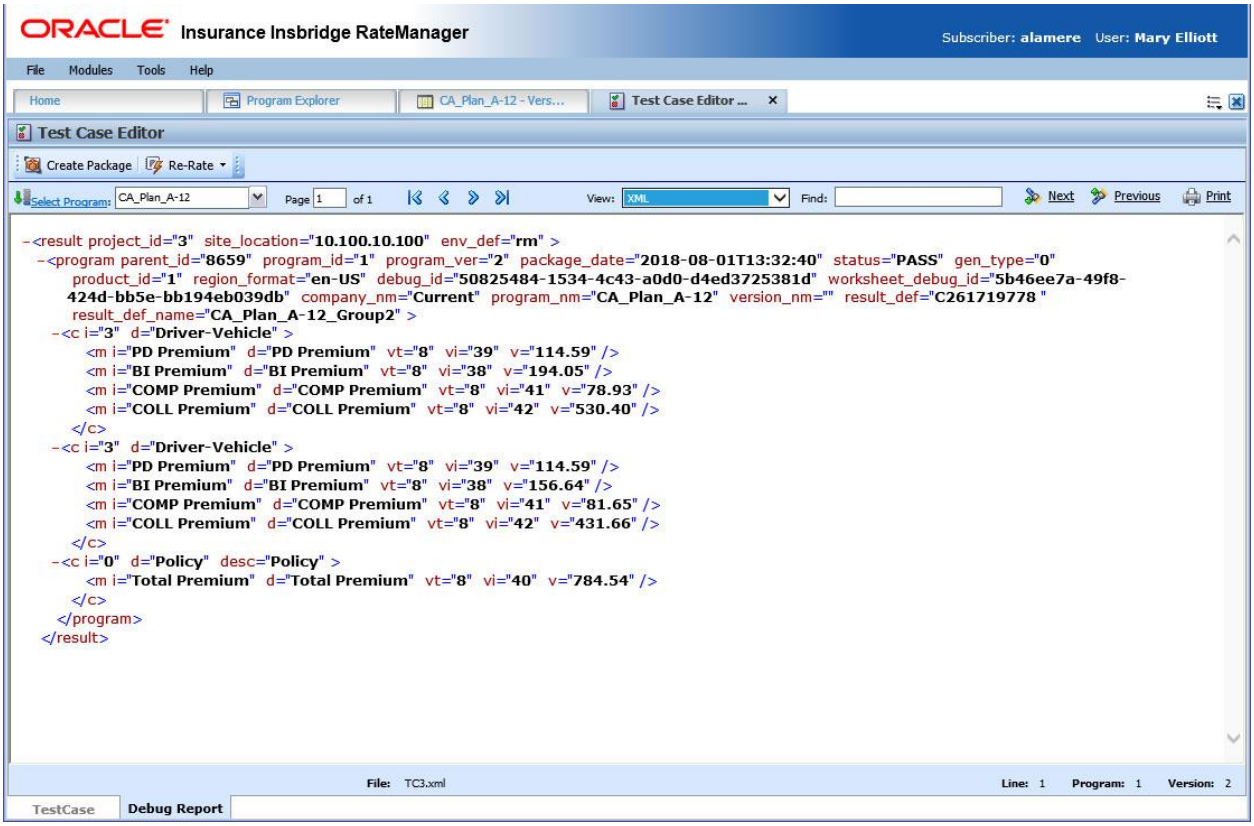


Figure 554 View Debug Reports XML

NOTE: *The date format used by Test Case Editor uses the format YYYY-MM-DD. This may not reflect the date format used by the program.*

Worksheet View

If worksheets are allowed, then Worksheet XML is an option in the View menu. If you want a worksheet available in Test Case Editor, it must be activated by a system administrator in the Preferences area. The default is for the worksheet option to be set to false. Most of the time, worksheets are not needed or required when testing a program under development.

There are two worksheet options, worksheet XML and sample worksheet report. Worksheet XML presents the XML details of the worksheet. The sample worksheet report option presents a sample of the worksheet report. Please see worksheets for more details on obtaining a complete worksheet report.

NOTE: *If information does not populate in the worksheet or sample worksheet, please make sure the worksheet is assigned to the Output Group being used.*

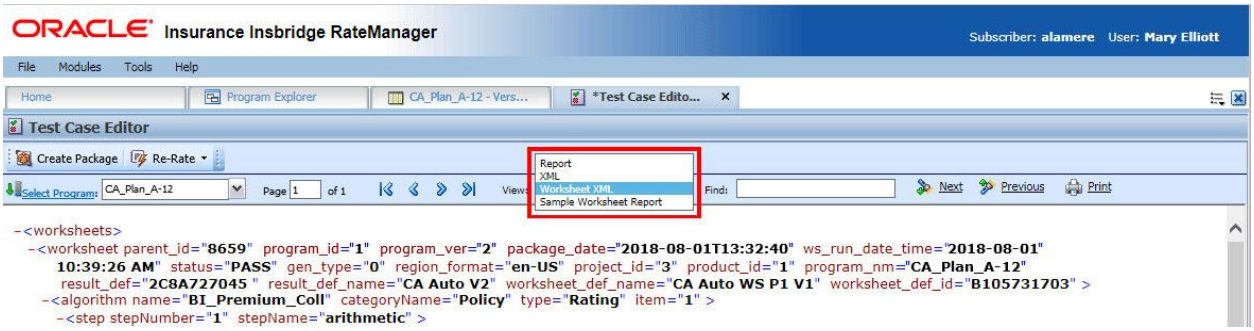


Figure 555 Worksheet Option in Test Case Editor

If the worksheet XML option is not listed then the option to view worksheets in Test Case Editor has not been enabled in Preferences. Contact your system administrator if you need worksheets in Test Case Editor.

Sample Worksheet

The sample worksheet option presents a sample of the worksheet.

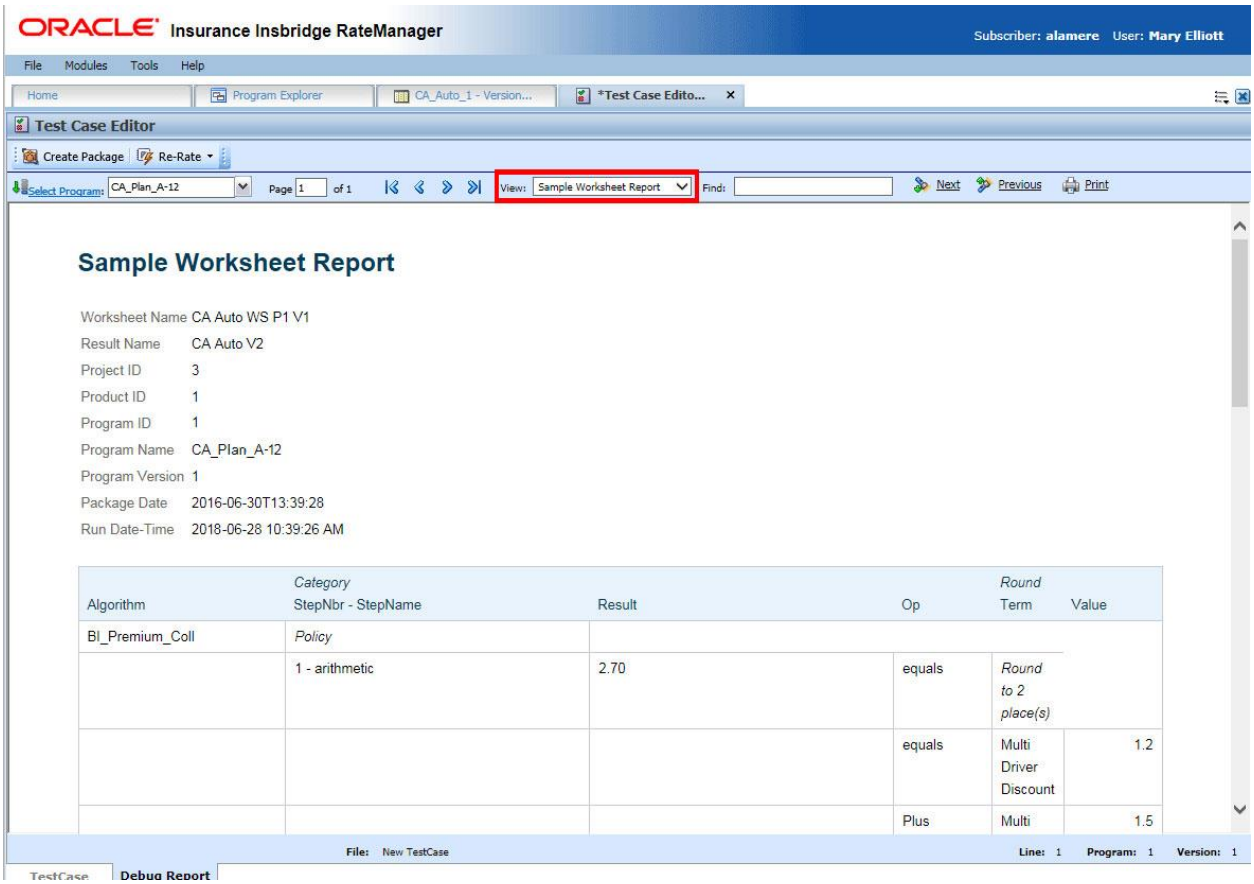


Figure 556 Sample Worksheet Report

The sample report lists out the steps taken and the elements that are used. It is clearly marked at the top of the report as Sample and should be used as a guide and not as a definitive report. Worksheet reports are available in Output mapping.

Message Templates in Debug Report

Message templates are displayed in the step where the message template was used. On the debug report,

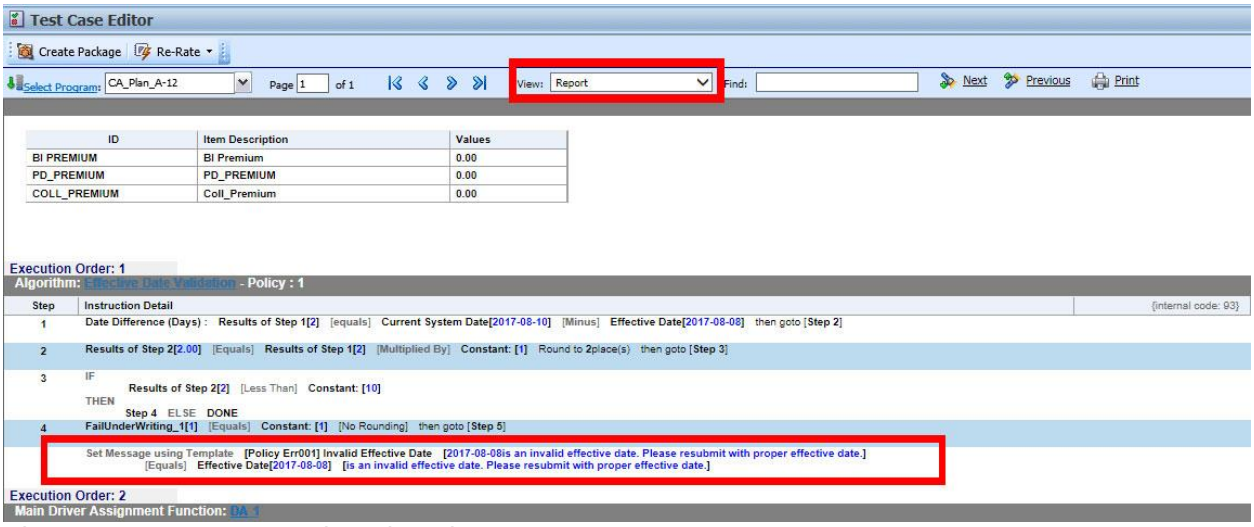


Figure 557 Message Templates in Debug Report

If the step is not run, the message is not displayed.

Message templates are displayed in the Messages node on the debug report in XML view.

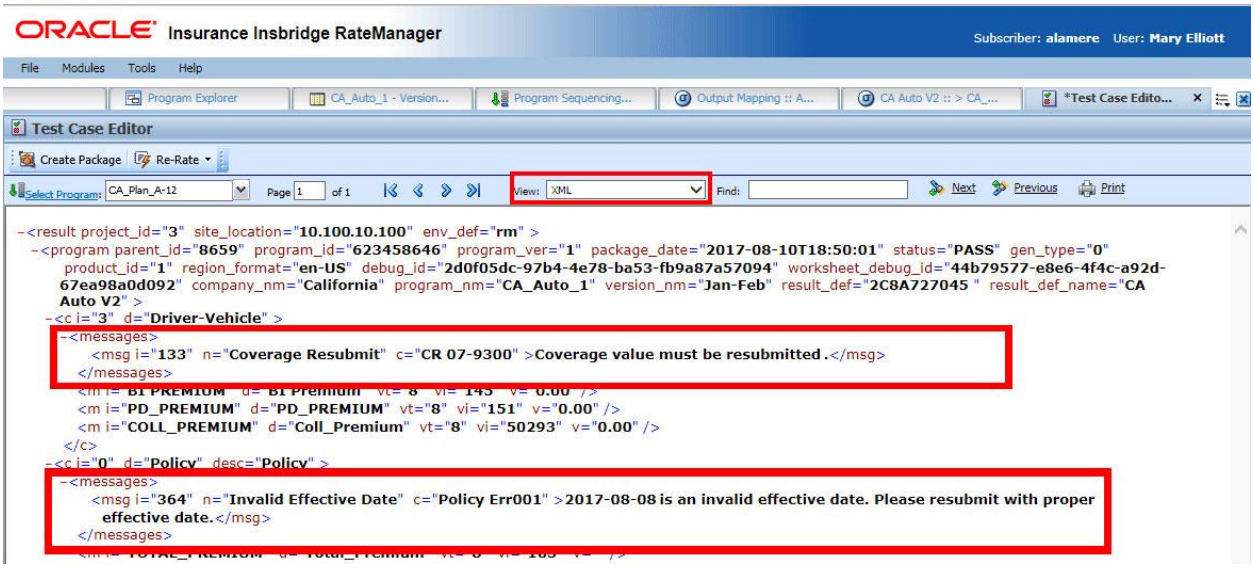


Figure 558 Message Templates in XML Debug Report

Signed and Unsigned Values in Debug Report

Date Difference steps can return a designation for a negative value. On the debug report, the designation is displayed in the step details. When a date difference step is signed, a sign is displayed only when the value returned in negative.

Figure 559 Signed Values in Debug Report

In the summary area of the debug report, signed date difference steps will display a negative value

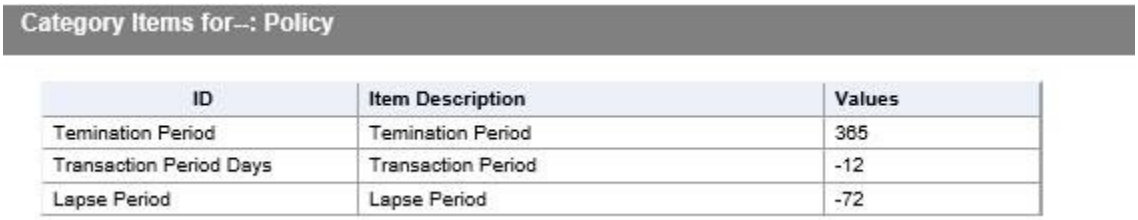


Figure 560 Signed Values for Category Items

Opening an Existing Test Case

You can open a saved test case by clicking the Open/Process New File option in the top bar menu. This will bring up a dialog window where you can select to open a test case associated with this program version or one saved to your local machine or network.

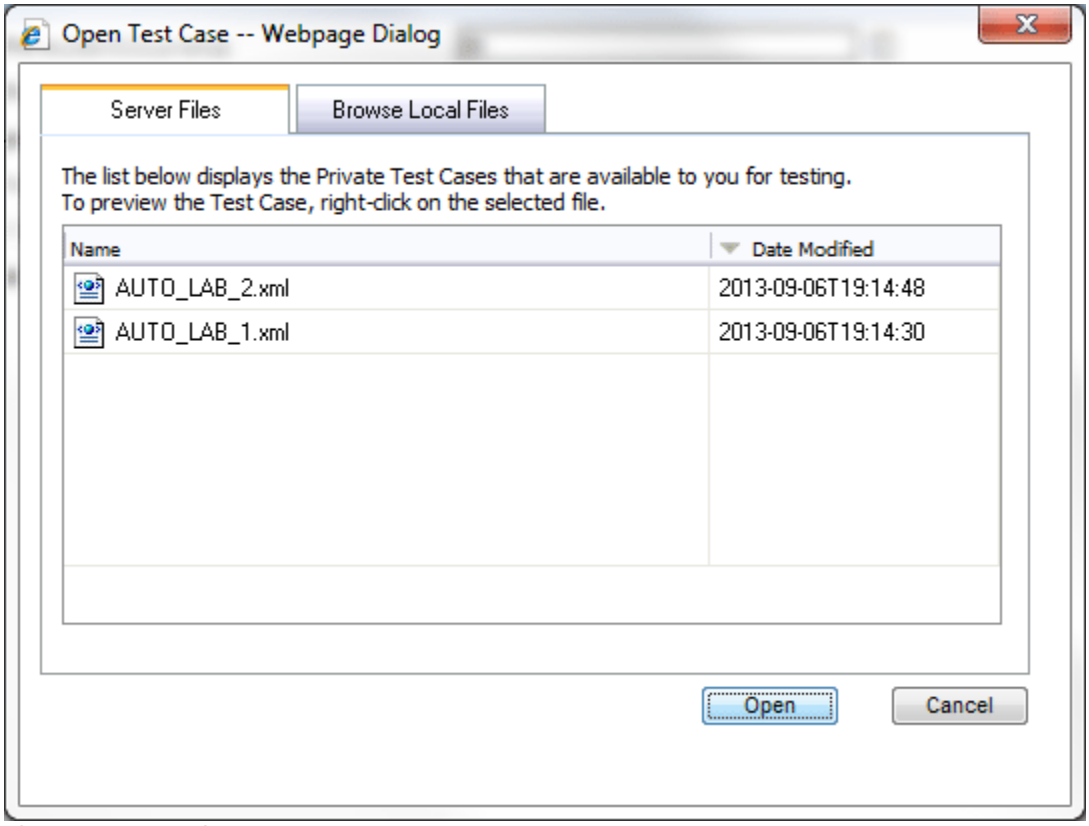


Figure 561 Opening a Test Case

There are two tabs:

- **Server Files:** These are the test cases associated with this program version. Test case files saved within the Test Program tab are saved to server files.
- **Browse Local Files:** These are the files saved outside of the Test Program tab to your local machine or network. You can test these files but you will not be able to save any changes. If you want to make changes to these files, please return to the source program or use the Testing Module.

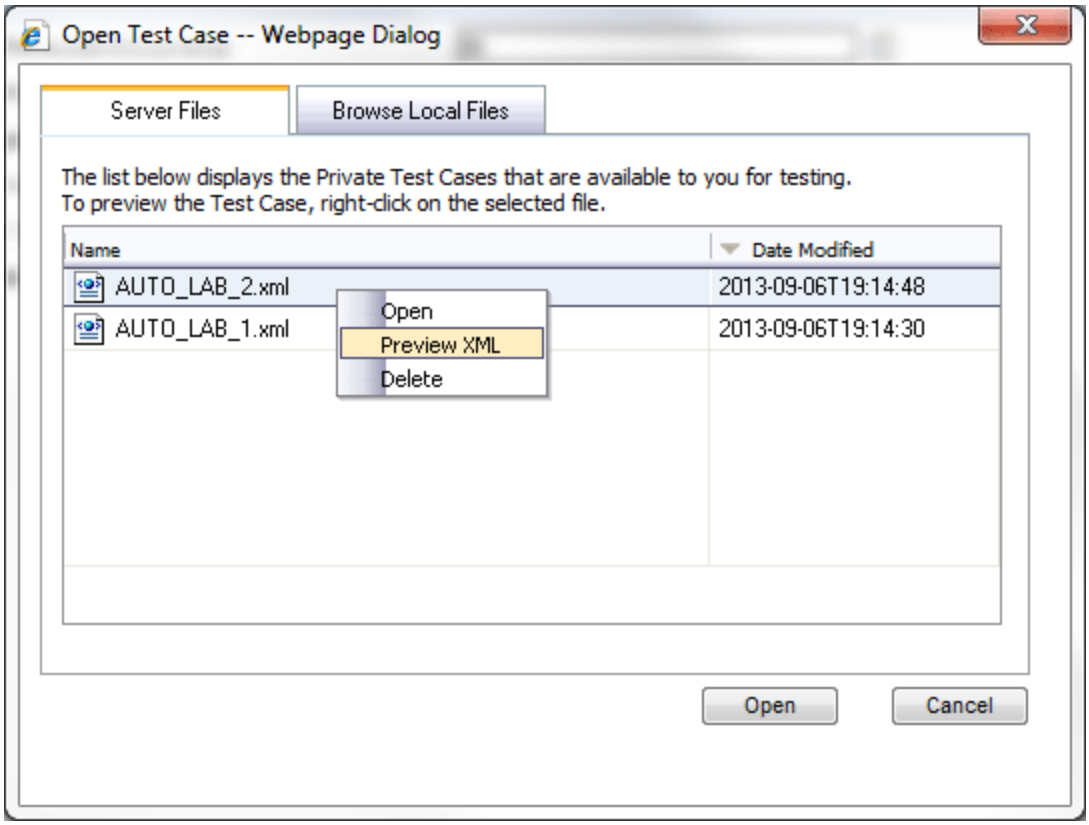


Figure 562 Right Click Menu for Opening Test Cases

Server files also have a right click menu. There are three options:

- **Open:** Opens the file in Test Program.
- **Preview XML:** Displays the XML in a separate screen. Preview allows for the XML file to be copied.
- **Delete: D**eletes the file. Deleting the file from here also deletes the file from the Testing module.

You cannot open a test case file outside of the version where the test case was saved. If you have created multiple test cases in version two of a program, you cannot open those test cases from version 4. You will have to enter version two or work in the Testing module.

Test Case Error

If there is a problem with the test case, an error message will be displayed at the top of the debug report. The error message “Unable to locate program version.” Will generally be displayed when you program is not packaged. To correct this error, package your program. You also can save this program without packaging and go to the Testing module. Unpackaged programs can be tested there.

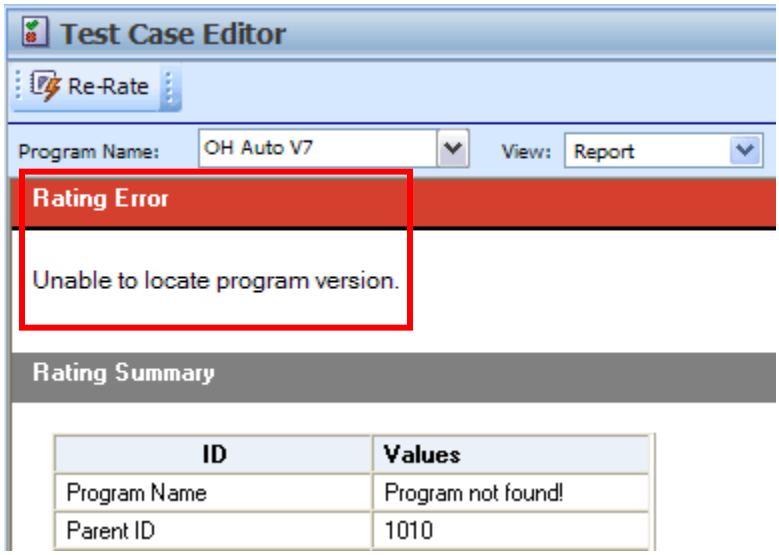


Figure 563 Test Case Error

Browse Local Files

You can browse to a test case XML file saved to your local machine or network.

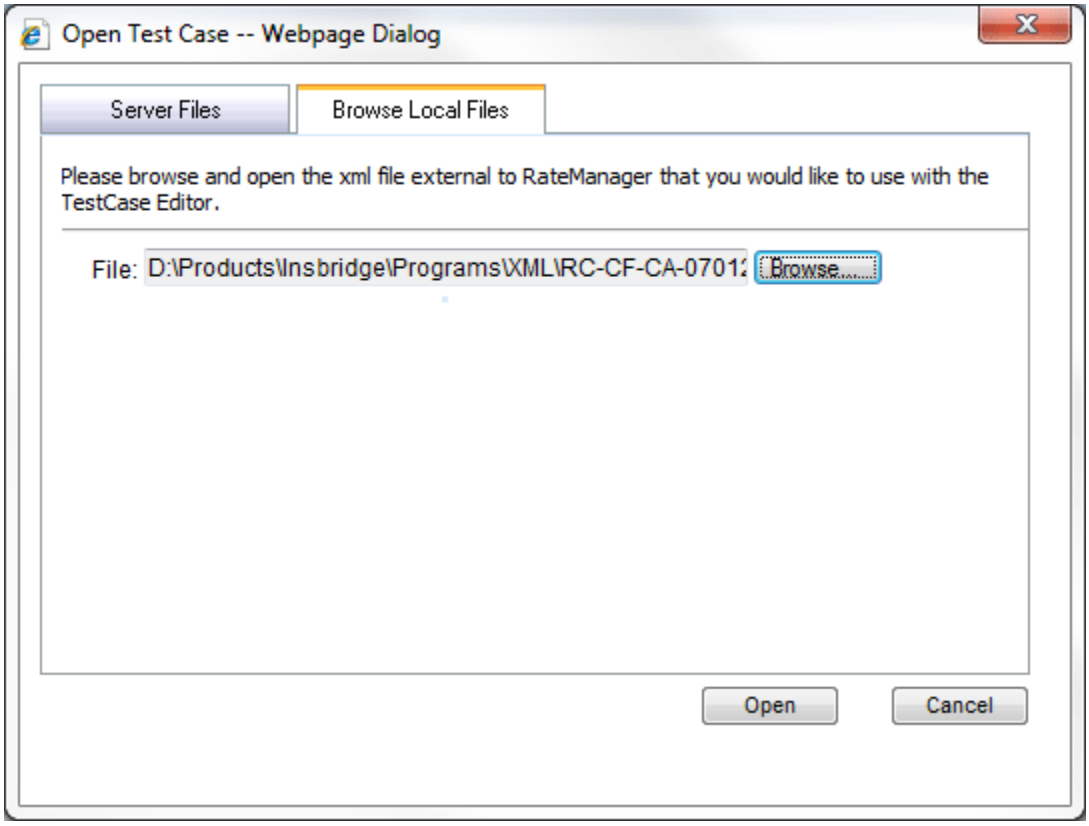


Figure 564 Opening a Local Test Case

To Open a File:

1. Select the file you want.
2. Click **OPEN** to immediately open the file.

If this is not the screen you wanted, click Cancel to close the dialog box and return to the Test Program tab.

Chapter 30

SYSTEM REPORTS

RateManager has a wide variety of reports available at the element level and at the program level. The majority of report options can be accessed from a right click menu. Most elements and programs in RateManager have a right click menu that will list the report options available.

Input XML and Program Version XML are available from an option on the top bar menu of the Program Explorer tab.

- **Program Difference Report** compares and displays the differences between two program versions or deployed packages showing the mismatched elements side by side.
  - Right-click option when positioned on a selected program
  - From drop-down lists, select baseline and comparison program versions
  - Select Run

- **Program Version Report** provides a complete listing of the entire contents of the selected program version (often used to supplement rate filings).
  - Right-click option when positioned on a selected program version

- **Input XML Report** in XML format presents the XML schema consisting of all categories and inputs from all defined programs under the selected project.
  - Tool bar option on program explorer when positioned on a selected program version

- **Program Version XML Report** in XML format presents the XML schema consisting of all categories and inputs used by the selected programs version.
  - Tool bar option on program explorer when positioned on a selected program version

- **View Dependency Report** for variables and algorithms presents a tree view of all the associated variables on which the selected element depends, how they relate to one another and a drill-down detail on each variable.
  - Right-click option when positioned on a selected element

- **Full Dependency Report** for variables and algorithms used in a sequence. Presents a tree view of all the associated variables on which the selected element depends, how they relate to one another and a drill-down detail on each variable.
  - Button in top bar menu on the Sequence tab and Worksheet tab

- **View Dependency Report All Algorithms Above** for variables and algorithms in a sequence from a chosen point. Presents a tree view of all the associated variables on which the selected element depends, how they relate to one another and a drill-down detail on each variable.
  - Right-click option when positioned on a selected element

- **View History Report** for all elements displays a summary audit history of changes including the type of change, who performed the change and the date the change was made.
  - Right-click option when positioned on a selected element

- **View PDF Report** for any table variable, a complete listing of all table data in PDF format.
  - Right-click option when positioned on a selected element

- **View Quick Reports** provides a detailed view of a table variable, calculated variable or algorithm showing all of the table properties or formula steps.

  - Right-click option when positioned on a selected element

- **View Quick Report (All Data)** shows all table data associated with the variable. This option is only available on table variables.

- **Table Difference Report** shows the differences between table data in selected revisions. This option is only available on table variables.

- **View Where Used Report** see if/where an element is used and in what variable, and variable revision.
  - Right-click option when positioned on a selected element

- **Audit Report** presents a detailed audit trail a history of all changes made within the RateManager designer. Report may be filtered by multiple criteria.
  - Menu selection under Tools\Audit
  - Tool bar selection to set filter options
  - Tool bar selection to run report

NOTE: *The Insbridge message service must be on for the reports to generate.*

PROGRAM DIFFERENCE REPORT

Program Difference Report compares two versions or packages from the same program and shows the differences between the two. When running a Program Difference Report:

- The versions and/or packages are from the same program. You will not be allowed to compare version 1 from program 1 to version 2 from program 2.
- If comparing two versions, the versions must be different. You cannot compare version 1 to version 1.
- Versions are current. The version you select will be the version currently in the system.
- Versions can be compared to a package. The package and the version can be from the same version. You can compare a version 1 package to the current version 1.
- Packages can be compared either to another package or to a current version.
- If comparing two packages, the packages must be different.
- Either RateManager or SoftRater packages can be used.

The differences that are recognized are:

- Elements that are used in one version or package but not used in the other.
- Changes to categories, inputs, variables, algorithms, sequences and result groups.

The comparisons are against elements that are used in the version or package. For example, if version 1 used revision 1 of an element and version 2 used revision 2, then the report would list that the element was used in both versions but with different revisions. Elements that are not used will not be listed on the report. For example, if an element was deleted and is no longer used by either version, the difference will not be noted.

This option is available on all programs in all s. To create a Program Difference Report, you must have either two versions, or one version and one package. If you have one version and no package, you will not be able to create a Program Difference Report.

Reports are displayed in the Insbridge Document Viewer. Reports can be saved to your local drive or network or printed. No editing can be done on the Insbridge Document Viewer.

NOTE: *OBI Publisher must be installed and running in order for Program and Program Version Reports to be displayed. If you receive an OBI Publisher error, please contact your system administrator.*

Running a Program Difference Report

1. From the Program Explorer tab, right click the program where you want to run a Program Difference Report.

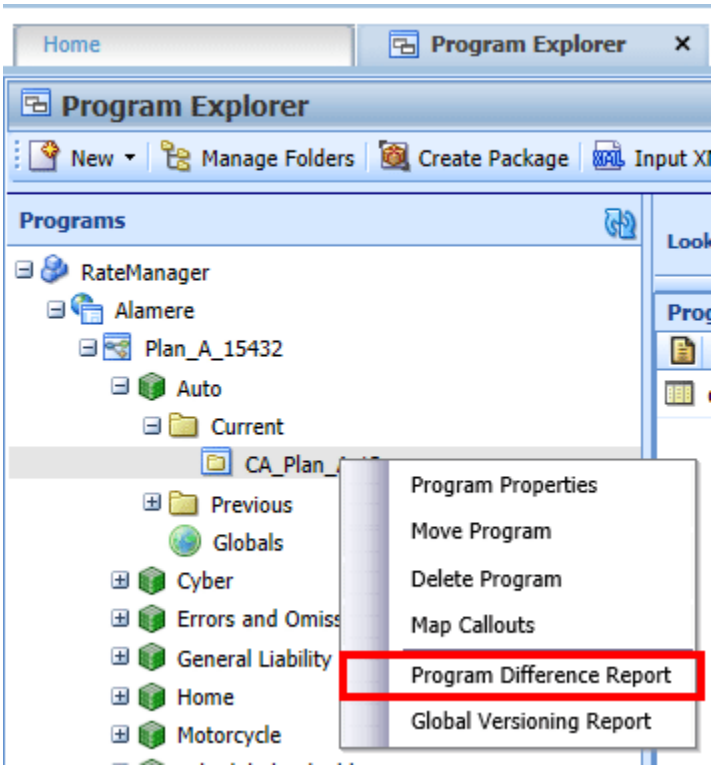


Figure 565 Program Difference Report Right Click Menu

2. Select **Program Difference Report** from the right click menu. A wizard is displayed.

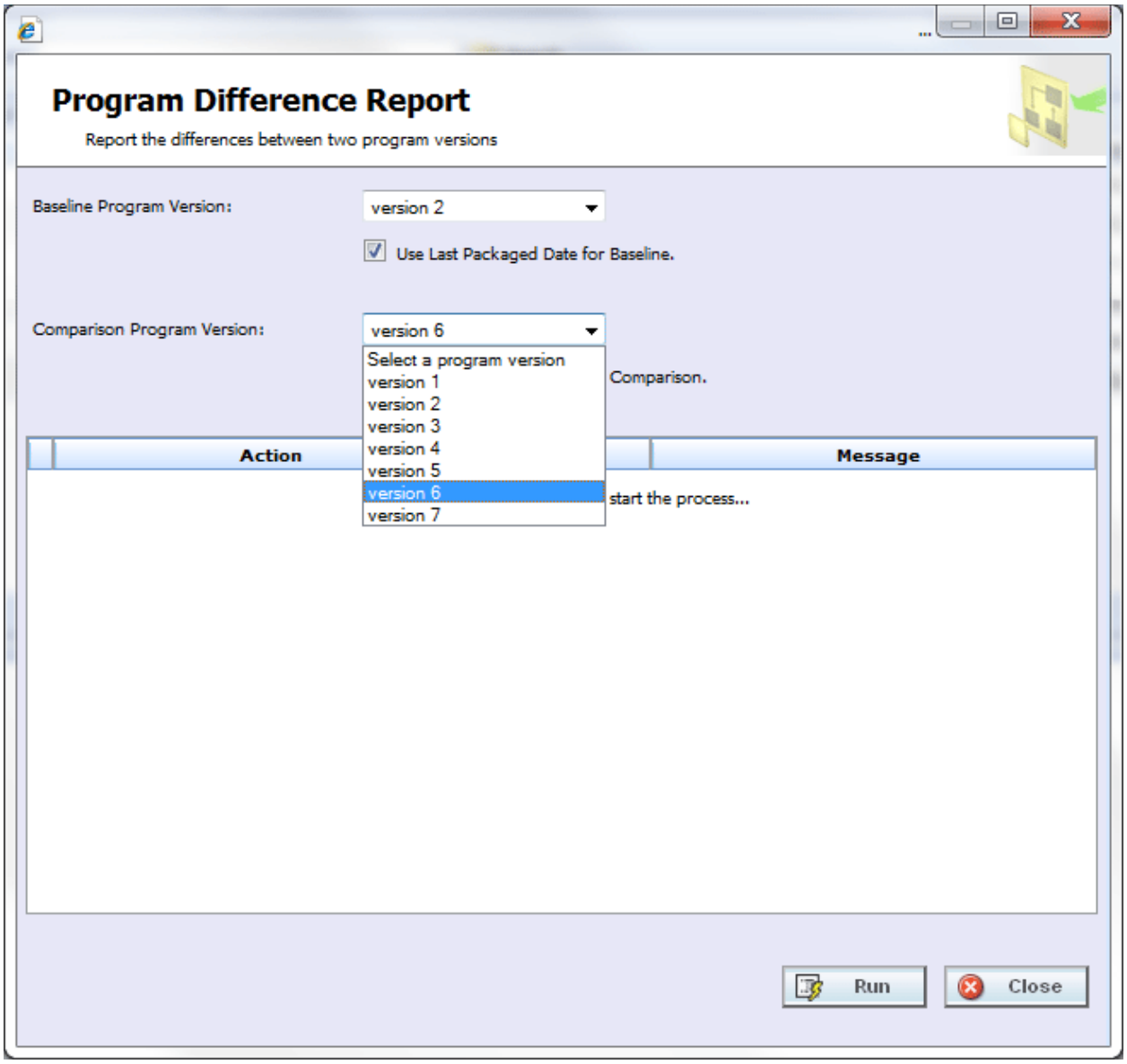


Figure 566 Selecting Programs Versions to Compare

3. Select the **Baseline Program Version**. All versions are listed. This version is the standard that the other version is compared against.

4. If the selected version has a package created, you have the option to check if you want to **use the last package** for this version. If the option is grayed out, no package was created.

5. Select the **Comparison Program Version**. All versions are listed. This can be any other version in the program, either newer or older.

6. If the selected version has a package created, you have the option to check if you want to **use the last package** for this version. If the option is grayed out, no package was created.

7. Click **Run** to start the report. The progress is displayed as the report runs.

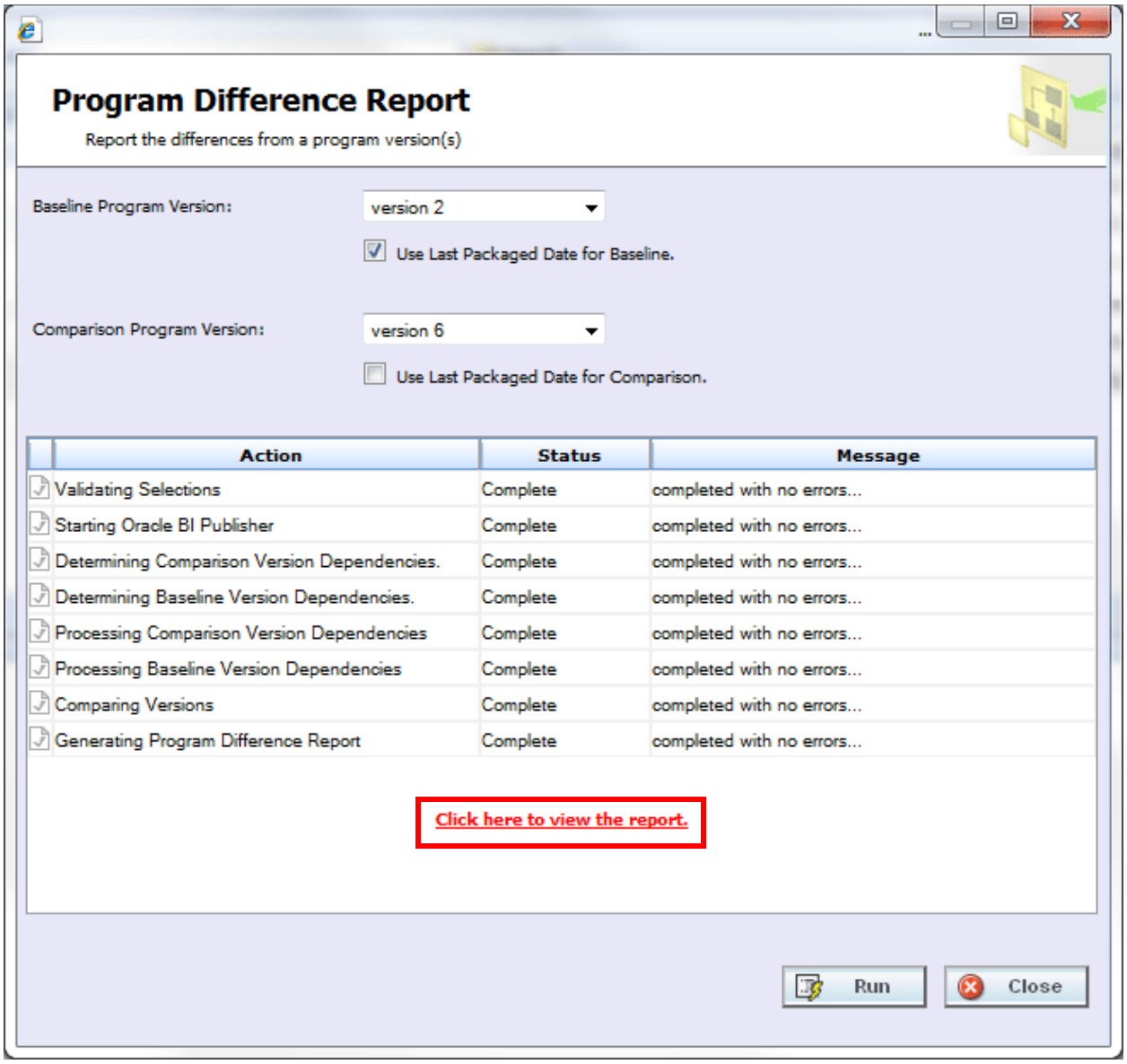


Figure 567 Completed Program Difference Report

8. Click the "**Click here to view the report**" link. The report is displayed in the Insbridge Document Viewer.

   If any errors occurred while running the report or displaying the report, please contact your system administrator.

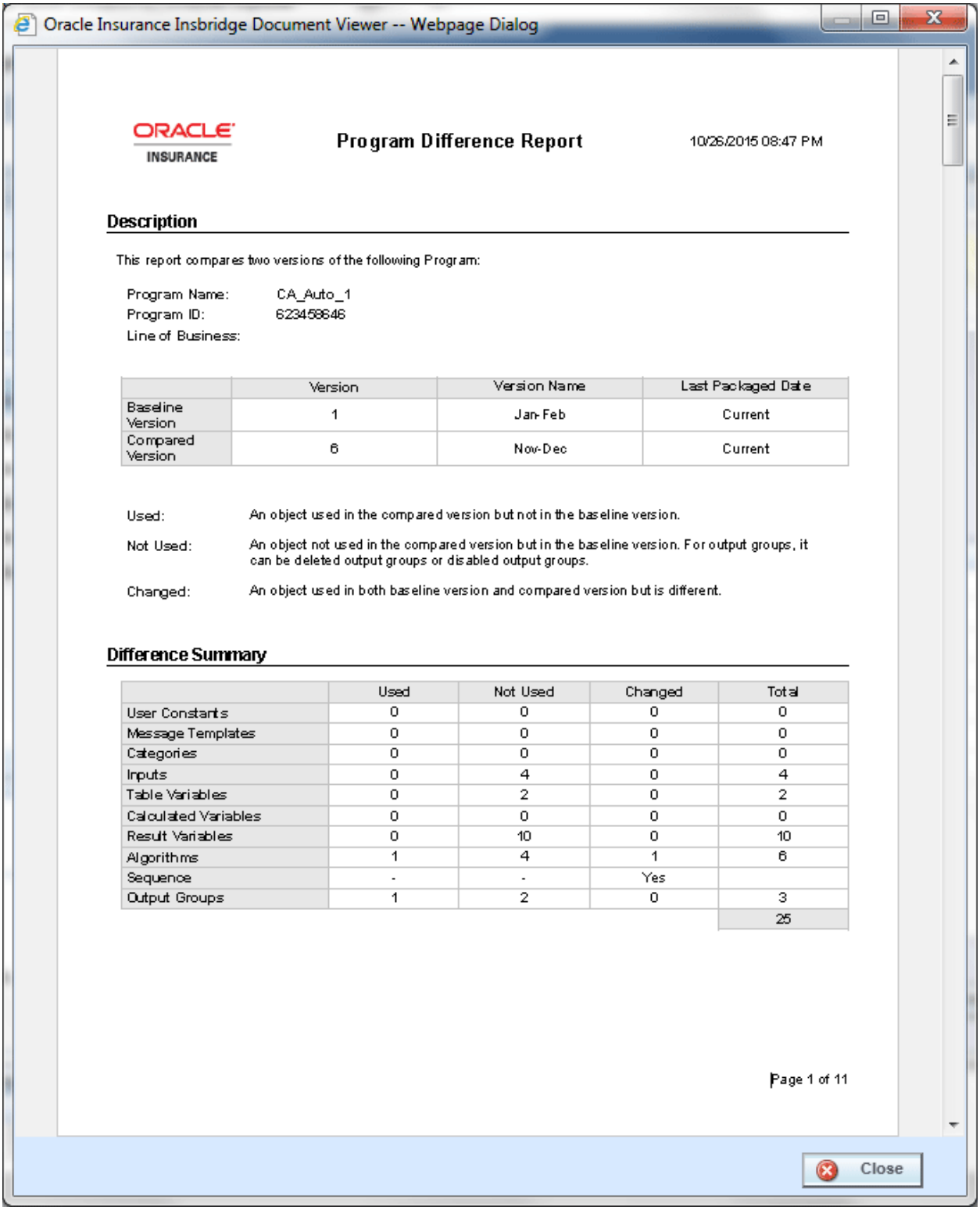


Figure 568 Program Difference Report

Reports are displayed in the Insbridge Document Viewer. These documents are read-only. Reports can be saved to your local hard drive or network or printed. No editing can be done on the Insbridge Document Viewer.

NOTE: *When you do a "Save As" from the Insbridge Document Viewer, you should rename the file. By default, when you go to save a file in the Insbridge Document Viewer, everything will be named "InsbridgeDocument". It is recommended that you rename the file.*

Report Details

The report has a description at the top, identifying the program, program ID and product. The Baseline and Compared versions are listed next. If the Last Packaged Date is **Current**, the program version was used. If there is a date listed, the last package for this version was used.

Definitions

- **Used:** an element that is used in one version but not in the other version. Used elements are listed underneath the version where they are found.
- **Not Used:** an object that is not used in one version but is in the other version. Not Used elements are represented by blank fields underneath the version where they are not found.
- **Changed:** an element that is used in both the baseline version and the compared version but has been changed in some way. Changes are displayed below the Not Used//Used section.

The following elements are shown in Used /Not Used section if they meet Used/Not Used criteria:

- **Categories**
- **Inputs**
- **Table Variables**
- **Calculated Variables**
- **Result Variables**
- **Algorithms:** if the name is different, it will show in the Not Used/Used section.
- **Output Groups:** deleted and disabled result groups will be counted as Not Used.

The following objects are shown in Changed section if any of the elements listed changes:

- **Categories:** Parent Category changes.
- **Inputs:** Data Type or Category changes.
- **Table Variables:** Data Type, Category, Default Value, Qualifiers or Data changes. Data changes list the dates of the last time the data was modified. The first date is for the baseline and the second is for the comparison. A change(s) was made some time between the two dates.
- **Calculated Variables:** Data Type, Category or Instructions changes.
- **Result Variables:** Data Type or Category changes.
- **Algorithms:** Category or Instructions changes. The step details will be listed out.
- **Sequence:** The order of Algorithms changes or one or more algorithm(s) used/not used.
- **Result Groups:** The combination of Description and Result_ID changes or Category or Enabled changes.

Only one report can be displayed in the Insbridge Document Viewer. You must close out the Insbridge Document Viewer to return to the report wizard. You can change the baseline or comparison program versions and run the report again. To return to RateManager, close the report wizard.

PROGRAM VERSION REPORT

A program version report provides the details of a program version.

1. From the Program Explorer tab, right click the program version where you want to run a Program Version Report.

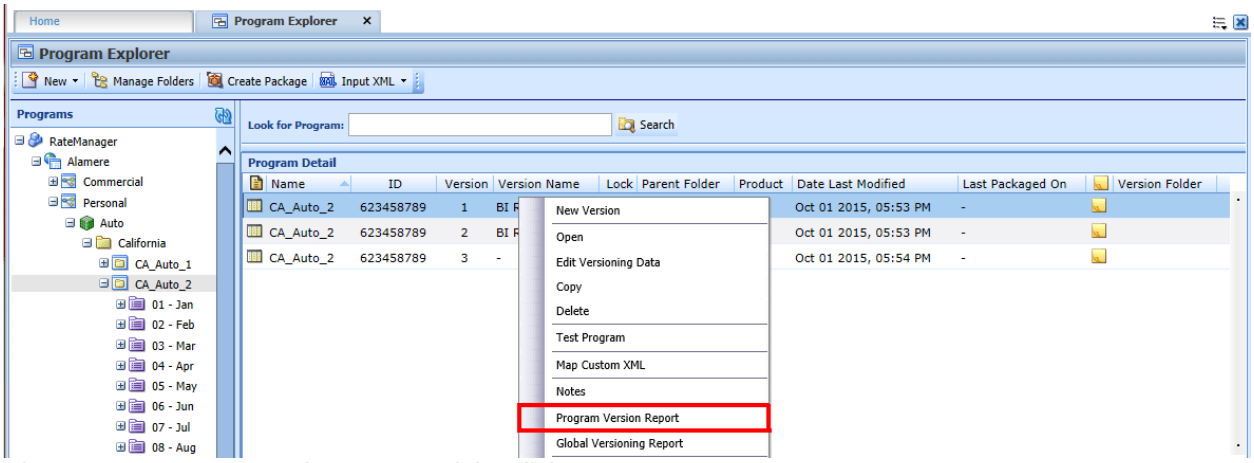


Figure 569 Program Version Report Right Click Menu

2. Select **Program Version Report** from the right click menu. The Insbridge Document Viewer is displayed. Your report will be generated in a moment.

Reports are displayed in the Insbridge Document Viewer. These documents are read-only. Reports can be saved to your local hard drive or network or printed. No editing can be done on the Insbridge Document Viewer.

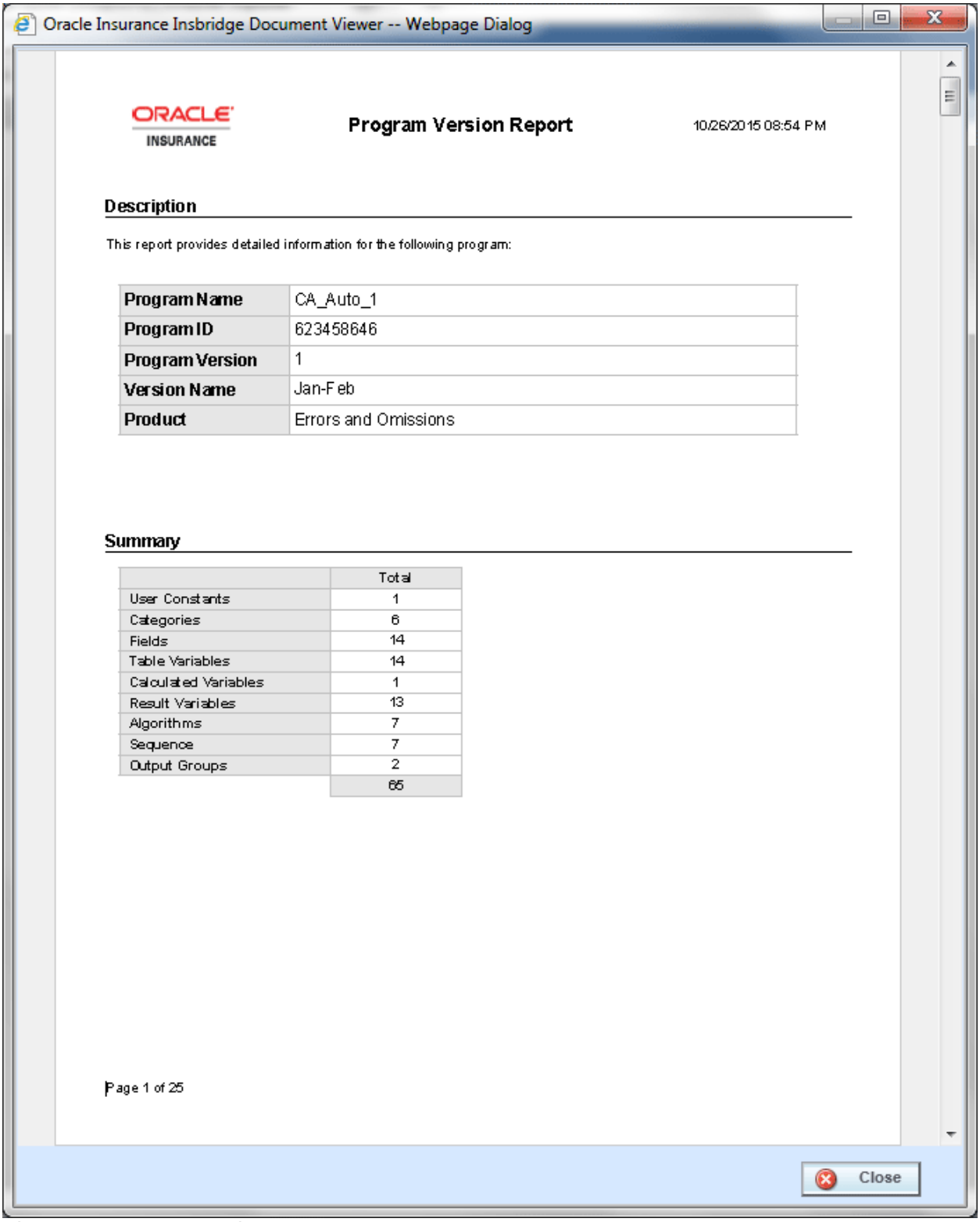


Figure 570 Program Version Report

NOTE: *When you do a "Save As" from the Insbridge Document Viewer, you should rename the file. By default, when you go to save a file in the Insbridge Document Viewer, everything is named "InsbridgeDocument". It is recommended that you rename the file.*

NOTE: *Report controls are displayed when the cursor is positioned in the lower portion of the report.*

Report Details

The report has a description at the top, identifying the program, program ID, program version and product. The Summary list is next. The Summary lists out the total number of elements found in the program version. The pages following detail all of the elements

Hyperlinks within the report allow you to jump to a named variable or other element. From the Description area on the front page of the report, you can link to the categories used or table variables used. In the algorithm details, there are links to the variables used.

When an element name in the report is clicked, you will be taken to the section of the program version report PDF document where the element is defined. Hyperlinks are highlighted with blue text and underline.

Program Version Report Hyperlinks

The hyperlinks are available from:

1. Table Variables Criteria
2. Data Source Variables Criteria
3. Calculated Variables Steps
4. Algorithm Steps
5. Sequence
6. Output Group

Hyperlinks are to these components:

1. Fields
2. Result Variables
3. Table Variables
4. Linked Variables
5. Data Source Variables
6. Calculated Variables
7. Constants
8. Message Templates
9. Algorithms

INPUTS XML

Selecting the **Inputs XML→Project** menu option displays the inputs XML for this project. You can view or copy the inputs XML. This feature can be accessed from the Program Explorer tab.

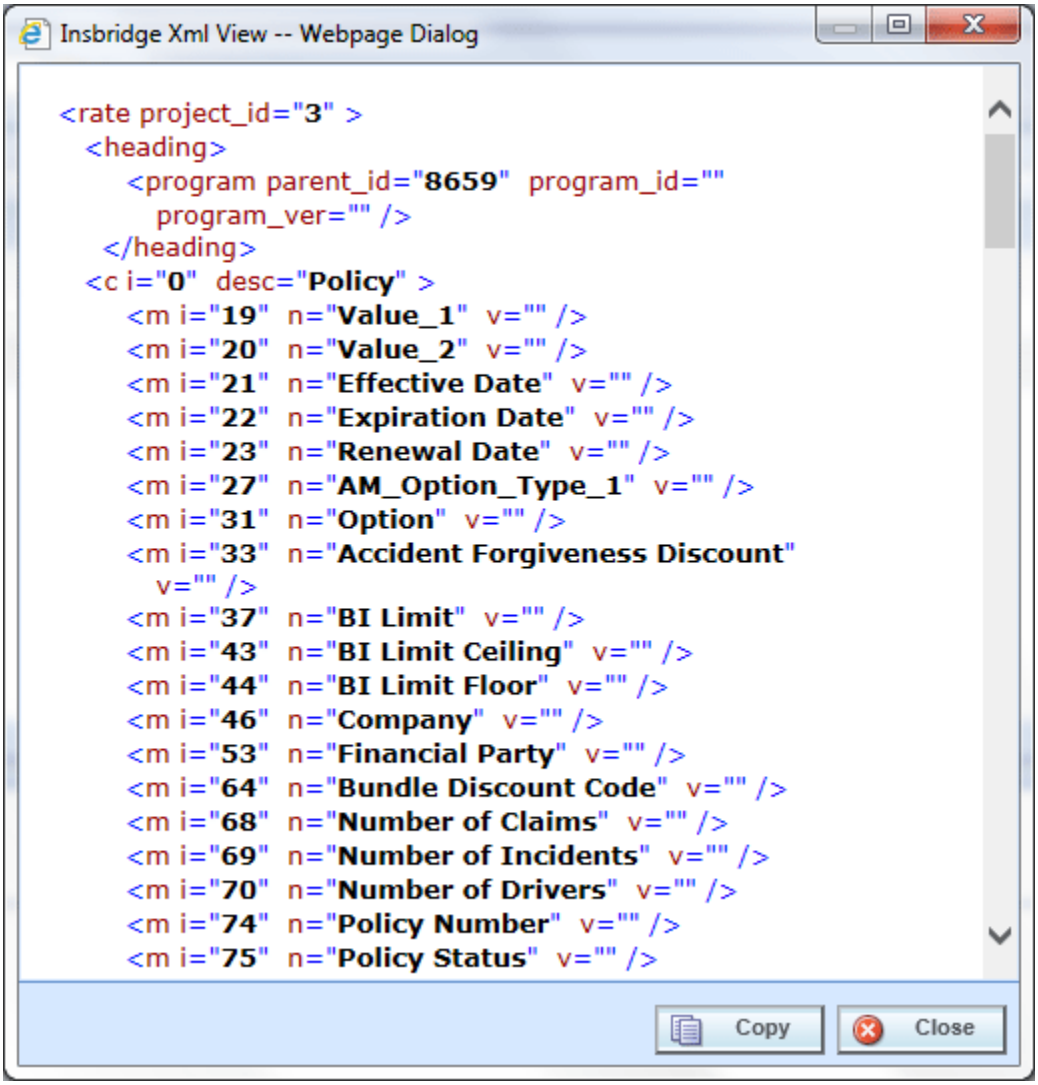


Figure 571 Inputs XML

To View Inputs XML

1. From the Program Explorer tab, select a project.
2. Select **Input XML→Project** from the top bar menu.
3. A separate window opens where you can view or copy XML for this project.
4. Click **Close** to close the window when you are finished.

PROGRAM INPUTS XML

Selecting the **Input XML→ Program Version** menu option displays the program inputs XML for the selected program version. You can view or copy the program inputs XML. This feature can be accessed from the Program Explorer tab.

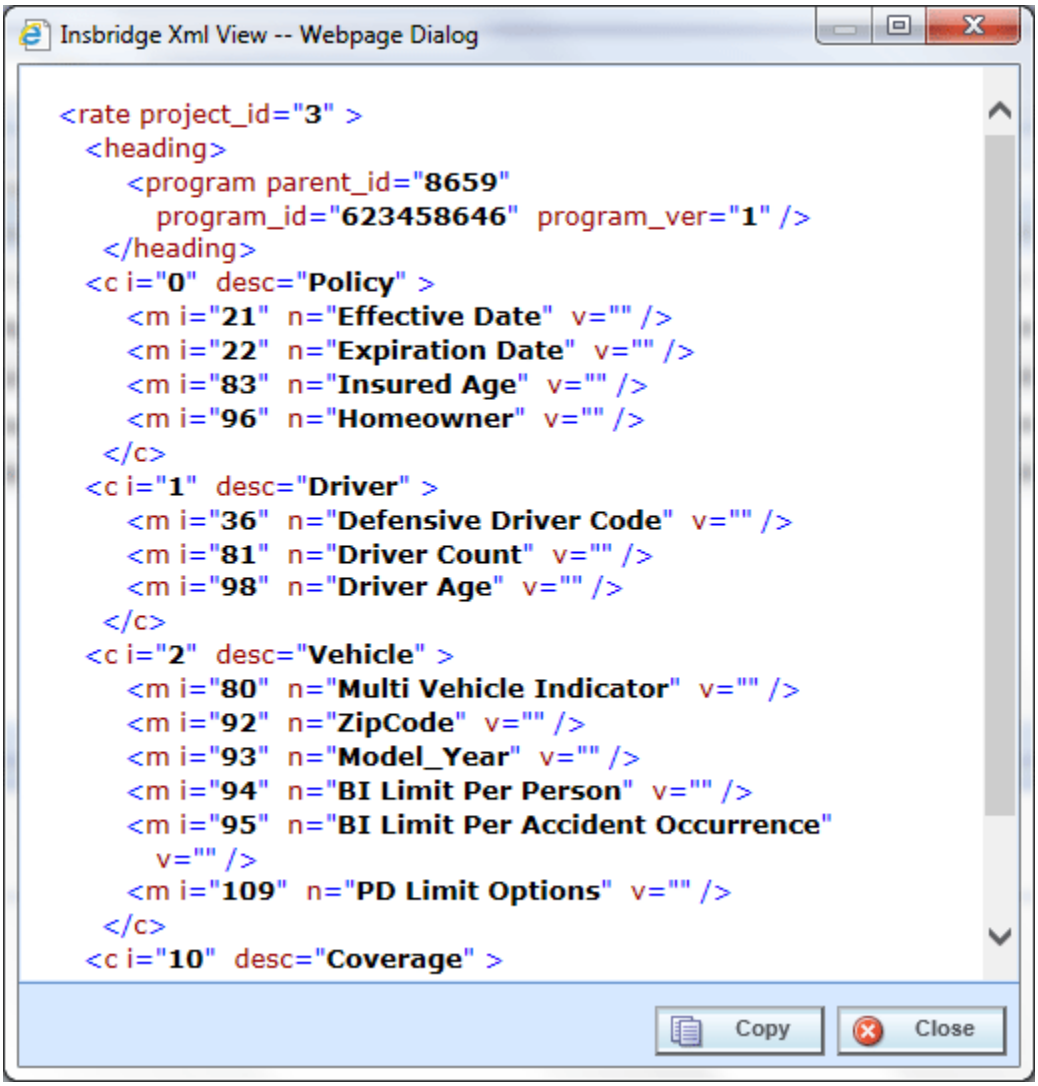


Figure 572 Program Inputs XML

Viewing Program Inputs XML

1. From the **Program Explorer** tab, select the domain type→domain→project→Folder→Program→Program Version.
2. Select **Inputs XML→Program Version** from the top bar menu.
3. A separate window opens where you can view or copy the XML for this program version.
4. Click **Close** to close the window when you are finished.

VIEW DEPENDENCY REPORT

A View Dependency Report gives you a tree view of a selected variable and all the associated variables used within. If you select to view an algorithm, all the inputs, results, table, and calculated variables used in the algorithm will be listed on the report. Details can be viewed by clicking the plus sign at the front of the variable. Inputs are listed first, with tables and calculated variables next, and result variables last. Inputs are listed in XML ID order.

Selecting a component variable publishes a quick view report in the lower portion of the window. Quick view reports are available on calculated variables, algorithms, and drivers assignments. Inputs and results do not have a quick view report. The quick view report contains the program name, revision, and working category along with other information.

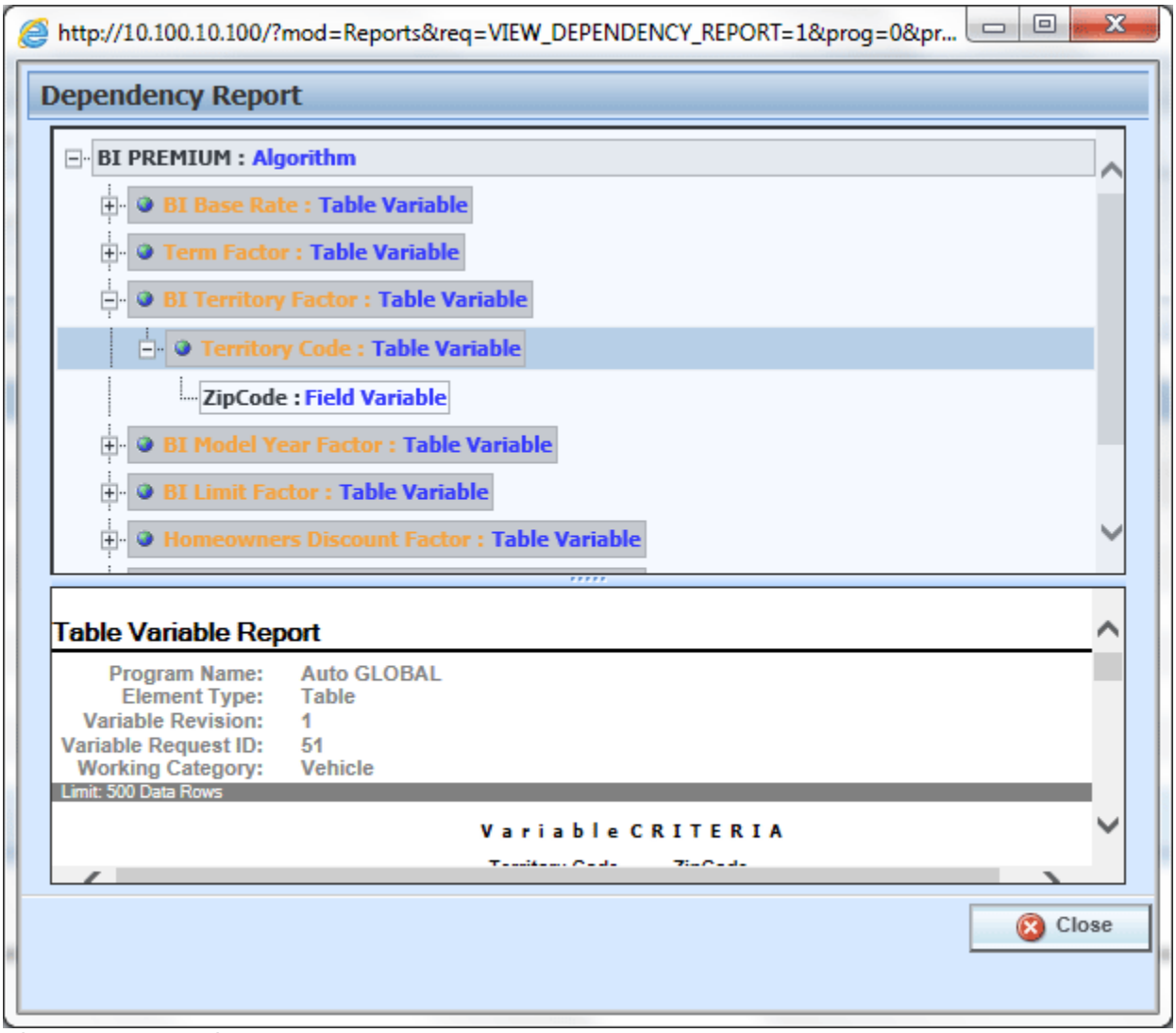


Figure 573 Dependency Report

- Yellow text indicates a table variable.
- Green text indicates a calculated variable.
- Variables that are boxed in dark gray indicate a global variable.
- Light gray boxes indicate a local variable.

Dependency reports are available on the right click menu of table variables, calculated variables, algorithms, and driver assignments. The View Dependency Report is not available on input variables and result variables.

FULL DEPENDENCY REPORT

A Full Dependency Report presents the relationships between variables in a sequence. You can view the algorithms, underwriting rules, and drivers assignments used in the sequence. Details regarding the variables used within these elements; calculated variables, tables, inputs, and results can be viewed by expanding the variable. A Full Dependency Report is useful when establishing or altering the execution sequence that is being used. Prior to making any changes, you can see what other elements may be affected by moving, adding, or deleting an algorithm. A Full Dependency Report is available on the Sequencing tab and Worksheet tab. Click the Full Dependency button in the top bar menu.

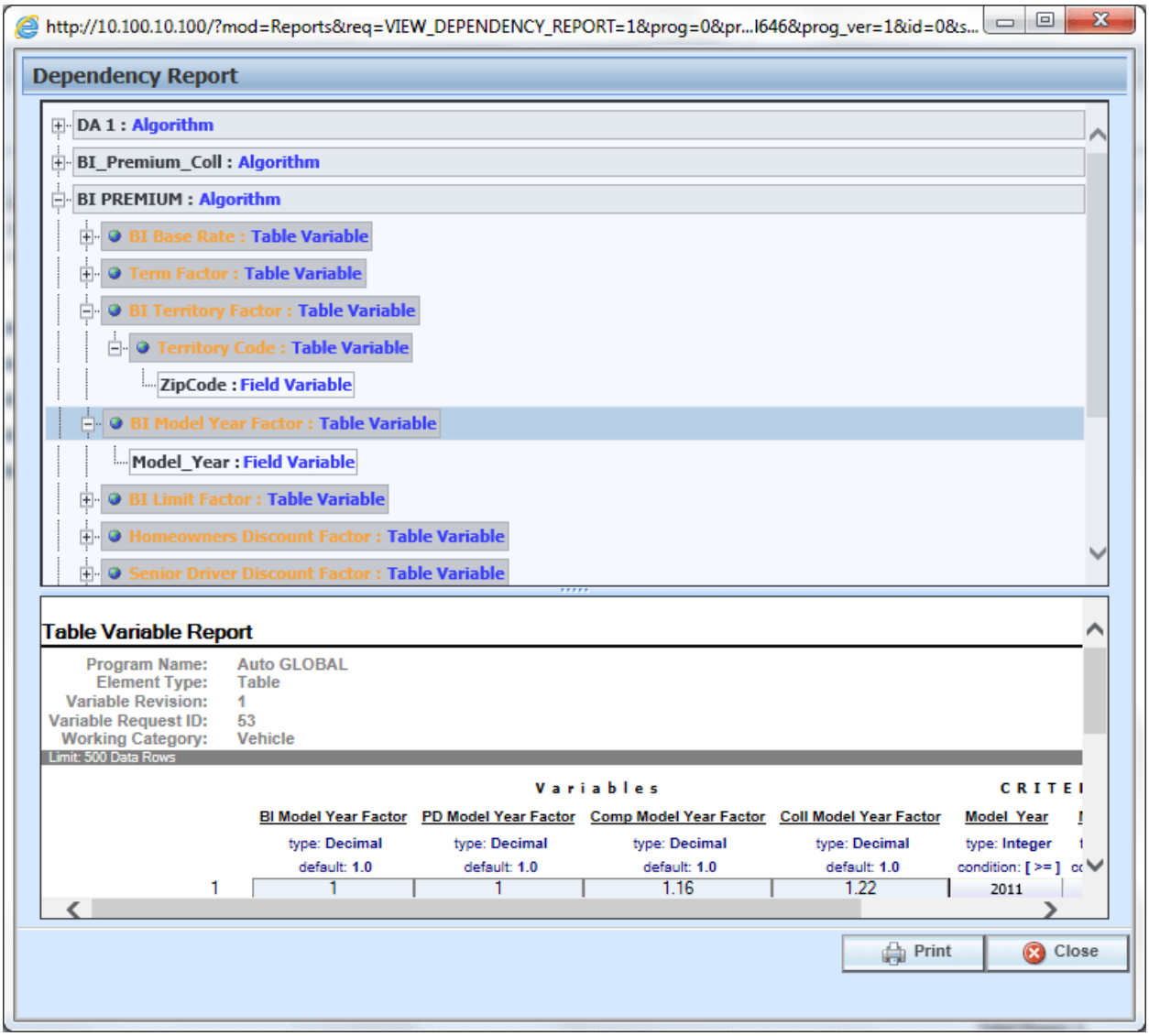


Figure 574 Full Dependency Report

A Full Dependency Report gives you a tree view of a selected sequence and contains all the algorithms, underwriting rules, and drivers assignments along with all the associated variables used in those elements. Details can be viewed by clicking the plus sign at the front of the variable.

Variables are listed in sequence order. When you select to view an algorithm, all the inputs, results, table, and calculated variables used in the algorithm will be listed. Inputs are listed first, with tables and calculated variables next, and result variables last. Inputs are listed in XML ID order.

Selecting a component variable publishes a quick view report in the lower portion of the window. Quick view reports are available on calculated variables, algorithms, and drivers assignments. Inputs and results do not have a quick view report. The quick view report contains the program name, revision, and working category along with other information.

VIEW DEPENDENCY REPORT ALL ALGORITHM ABOVE

A View Dependency Report All Algorithms Above Report presents the relationships between variables in a sequence from a chosen point. You can view the algorithms, underwriting rules, and drivers assignments used in the sequence from the selected point and above. Details regarding the variables used within these elements; calculated variables, tables, inputs, and results can be viewed by expanding the variable.

A View Dependency Report All Algorithm Above Report is useful when you want to view only the relationships prior to the point where you want to make a change. By selecting the exact algorithm you want to change, you can see what other elements may be affected by the change. A View Dependency Report All Algorithm Above Report is available on the right click menu of any algorithm, underwriting rule, or driver assignment used in the sequence on the Sequencing tab only. Select the algorithm, underwriting rule, or driver assignment where you want to view the variables above it. Right click and select View Dependency Report All Algorithm Above from the menu.

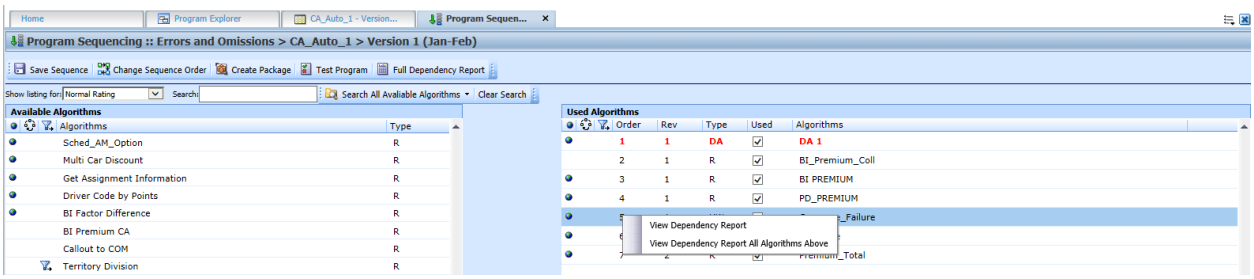


Figure 575 View Dependency Report All Algorithm Above

Variables are listed in sequence order. When you select to view an algorithm, all the inputs, results, table, and calculated variables used in the algorithm will be listed. Inputs are listed first, with tables and calculated variables next, and result variables last. Inputs are listed in XML ID order.

Selecting a component variable publishes a quick view report in the lower portion of the window. Quick view reports are available on calculated variables, algorithms, and drivers assignments. Inputs and results do not have a quick view report. The quick view report contains the program name, revision, and working category along with other information.

VIEW HISTORY

View History will display a detailed account of the action taken, who performed the action, the date the action happened, program information and any element information. View History is an optional feature that must be enabled by the system administrator on the User Groups tab. If auditing is set to Normal or High, actions will be recorded and presented in a View History report. If auditing is not enabled, the View History report will not display any actions.

The top portion of the screen will list the actions taken with the most current listed at the top. To view the details of the action, click **SHOW PREVIEW WINDOW** in the lower right corner of the screen and then click an action you want to view. The details will be displayed in the lower portion of the screen.

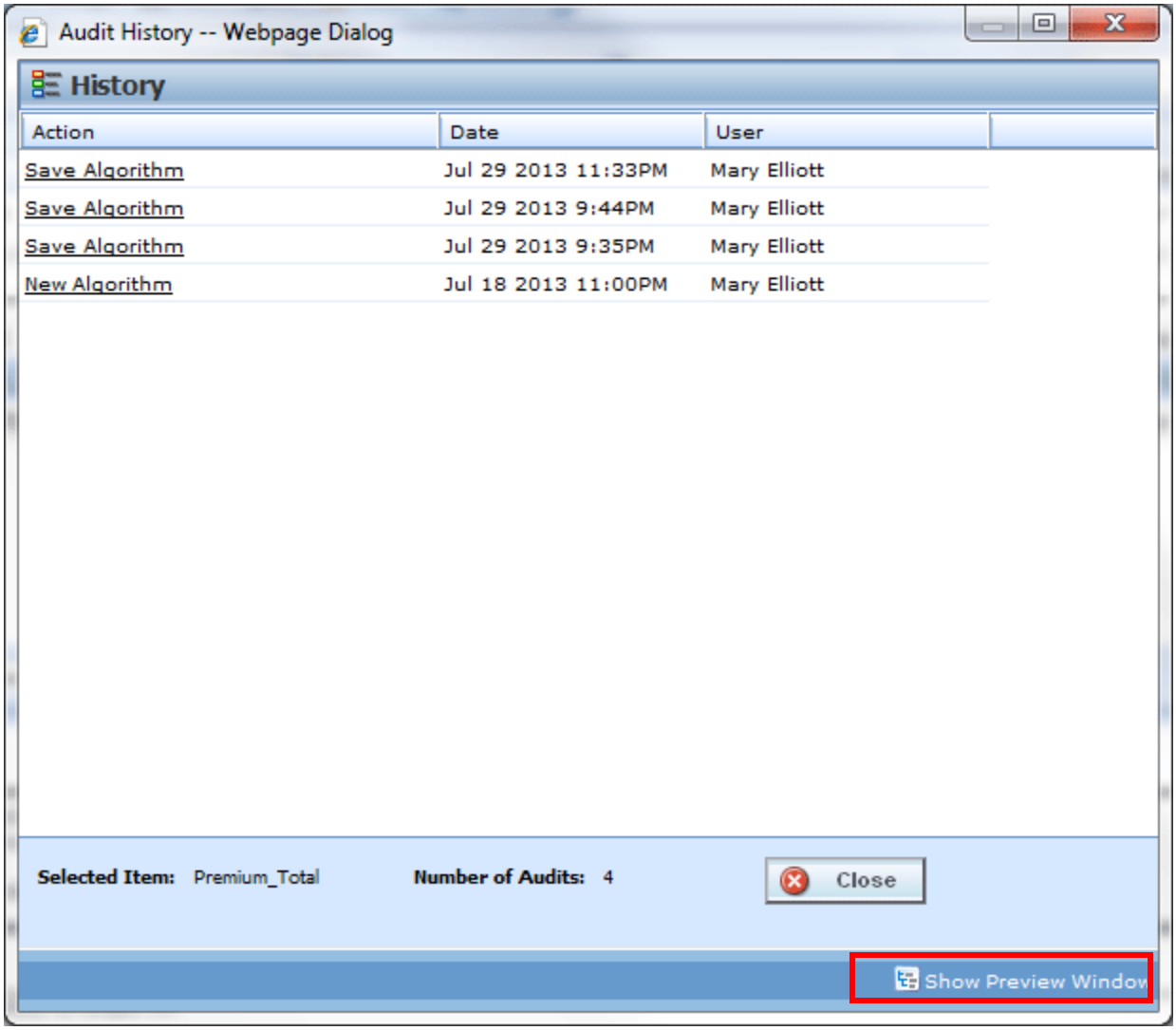


Figure 576 View History

If you do not want to view details, click **HIDE PREVIEW WINDOW** to remove the preview window.

View History is available on inputs, result, table and calculated variables, algorithms, and driver assignments from the right click menu.

VIEW PDF REPORT

Any table variable, either global or local, can have a PDF of the data table created. The report displays 10,000 rows at a time. If there are more than 10,000 rows, multiple reports will be produced; each one containing no more than 10,000 rows. You can navigate through multiple reports using the buttons at the bottom of the window.

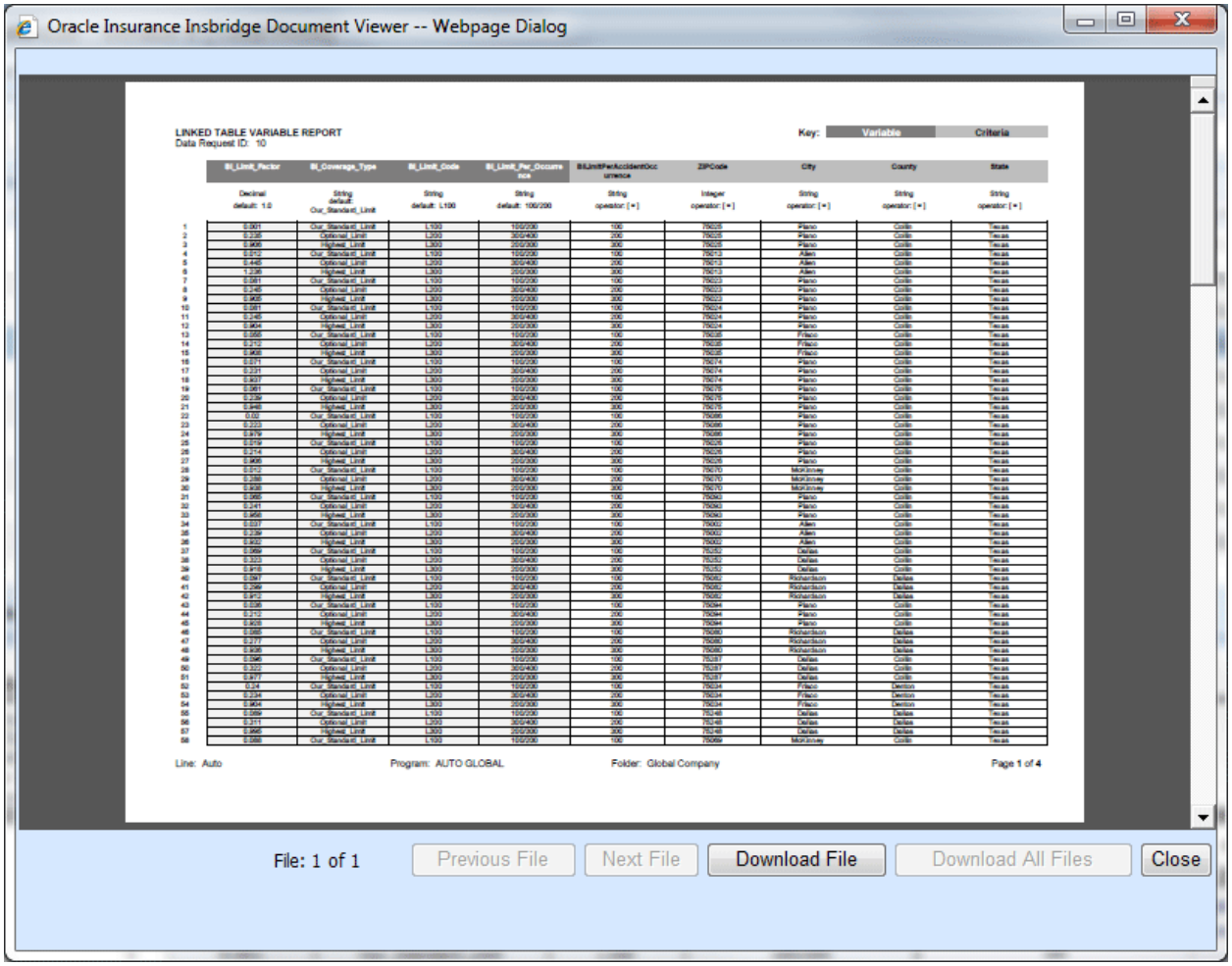


Figure 577 View PDF Report

NOTE: *PDF viewing software such as Acrobat Reader® is required to view PDF files.*

To View a PDF

1. Navigate to Program Version that contains the Table Variable you want to view.
2. On the Program Details tab, select the Table Variable where you want a PDF.
3. Right click and select **View PDF Report**.
4. The PDF opens in a separate window. You can Print or Save to your local machine or network if needed.

Viewing Options

There are buttons located on the lower portion of the window that allow you to view and download report files, and close the window. Previous File, Next File, and Download All Files buttons are not displayed on reports with less than 10,000 rows. Only large table variables with over 10,000 rows that produce multiple reports have these options displayed.

If the button is grayed out, the option is not available.

Previous File: Returns to the previous file view.

Next File: Opens the next file.

Download File: Downloads the PDF file to your local machine or network. The Download File option does not download the table data.

Download All Files: Downloads all the PDF files to your local machine or network. The Download All Files option does not download the table data.

Close: Closes the window.

VIEW QUICK REPORT

View Quick Reports give a detailed view of a variable, algorithm or driver assignment showing either all of the steps or all of the table data, depending on type.

These reports are useful when reviewing because it eliminates the need to click through each step. Additionally, viewing a Report when in Edit Mode reduces the need to search for and review the variable, algorithm or scenario outside of Edit Mode.

Viewing a Report

There are two ways to view a report. The first is to use the **Preview Window** at the bottom portion of the screen. The preview window can be either shown or hidden. When shown, the preview window will update to show the currently selected variable, algorithm or scenario. To hide the preview window, click Hide Preview Window.

The other option is to right click a variable, algorithm or driver assignment and select **View Quick Report** from the menu.

The report will load in a separate screen and may be viewed or printed.

- **View Quick Report (All Data)**: This option is only available on table variables and shows all table data associated with the variable. The report returns a maximum of 10,000 rows.

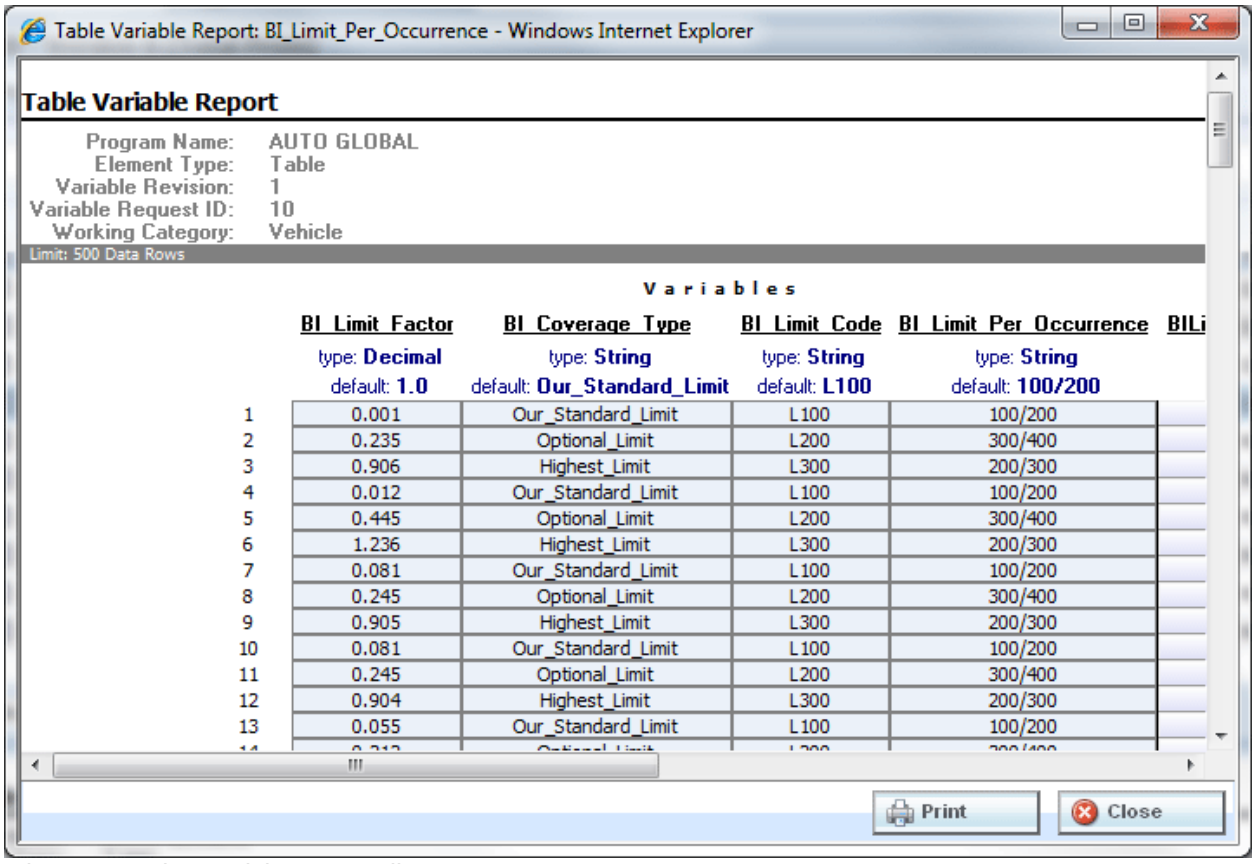


Figure 578 View Quick Report All Data

- **View Quick Report**: This option is for table and calculated variables, algorithms or scenarios. For calculated variables, algorithms, and scenarios, you will get a breakdown of steps and the variables used. For Table variables, the variables used and criteria will be displayed.

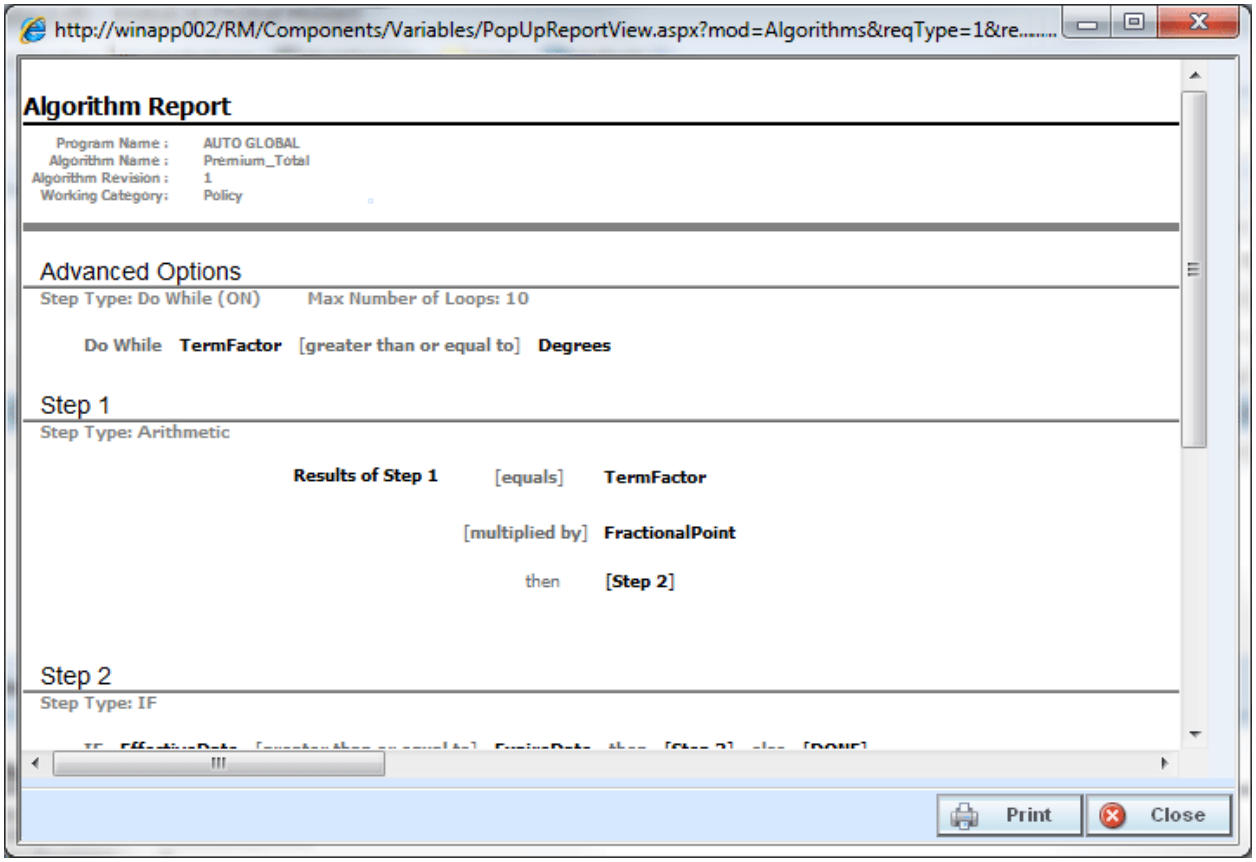


Figure 579 View Quick Report

VIEW REPORT

When you are in Edit mode for algorithms, calculated variables, table variables, and driver assignments, you can view the pertinent information for a result variable or an input variable by right clicking the variable. A window is displayed with the information you need, including Type and Category, allowing you to stay in Edit mode.

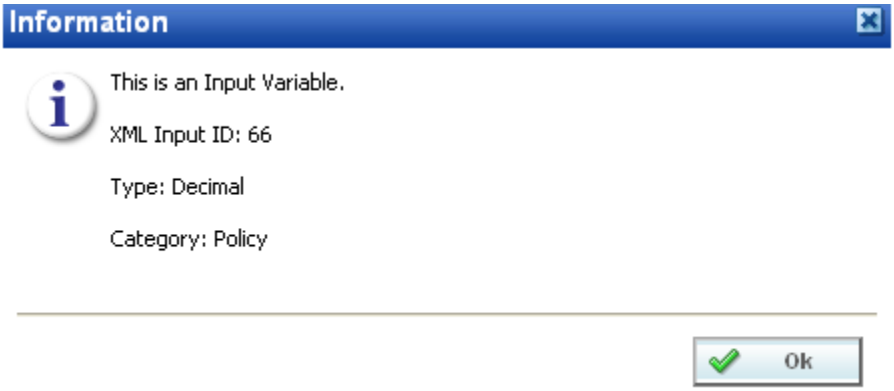


Figure 580 View Report

TABLE DIFFERENCE REPORT

Table Difference Report shows the differences between two selected table variable revisions. In a Table Difference Report one revision serves as the base to which the other revision will be compared. Differences are listed in a PDF report. The Table Difference Report is available to any global or local table variable. The differences that are recognized are:

- Table data that has a different value. The row and the differing values are displayed.
- Changes to attributes and columns.

Only revisions of the same table variable can be compared. You cannot compare revision 2 of Table A to revision 3 of Table B.

Reports are displayed in the Insbridge Document Viewer. Reports can be saved to your local drive or network or printed. No editing can be done on the Insbridge Document Viewer.

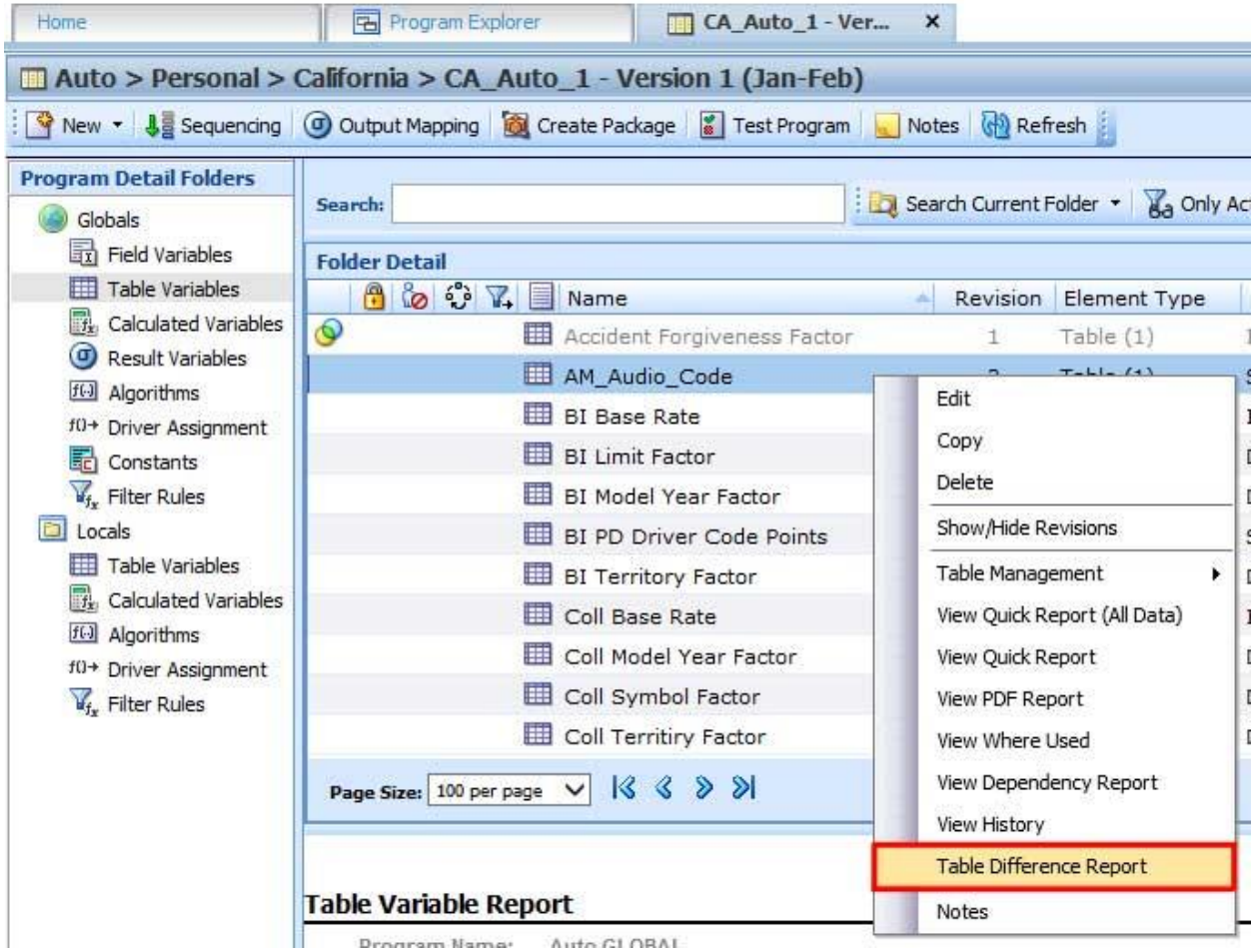


Figure 581 Table Difference Report Right Click Menu

NOTE: *OBI Publisher must be installed and running in order for reports to be displayed. If you receive an OBI Publisher error, please contact your system administrator.*

Where you select to run the report determines what revision acts as the base. You can select the comparison table only. If you run the report from the Program Details tab, the base revision is revision 1. If you want another revision to serve as base, you must show/hide revisions and select the revision you want to act as base. Right click and select to run the Table Difference Report from that revision.

Running a Table Difference Report

1. Select the table variable where you want to view the differences between revisions.
2. Right click and select **Table Difference Report**. A wizard is displayed.

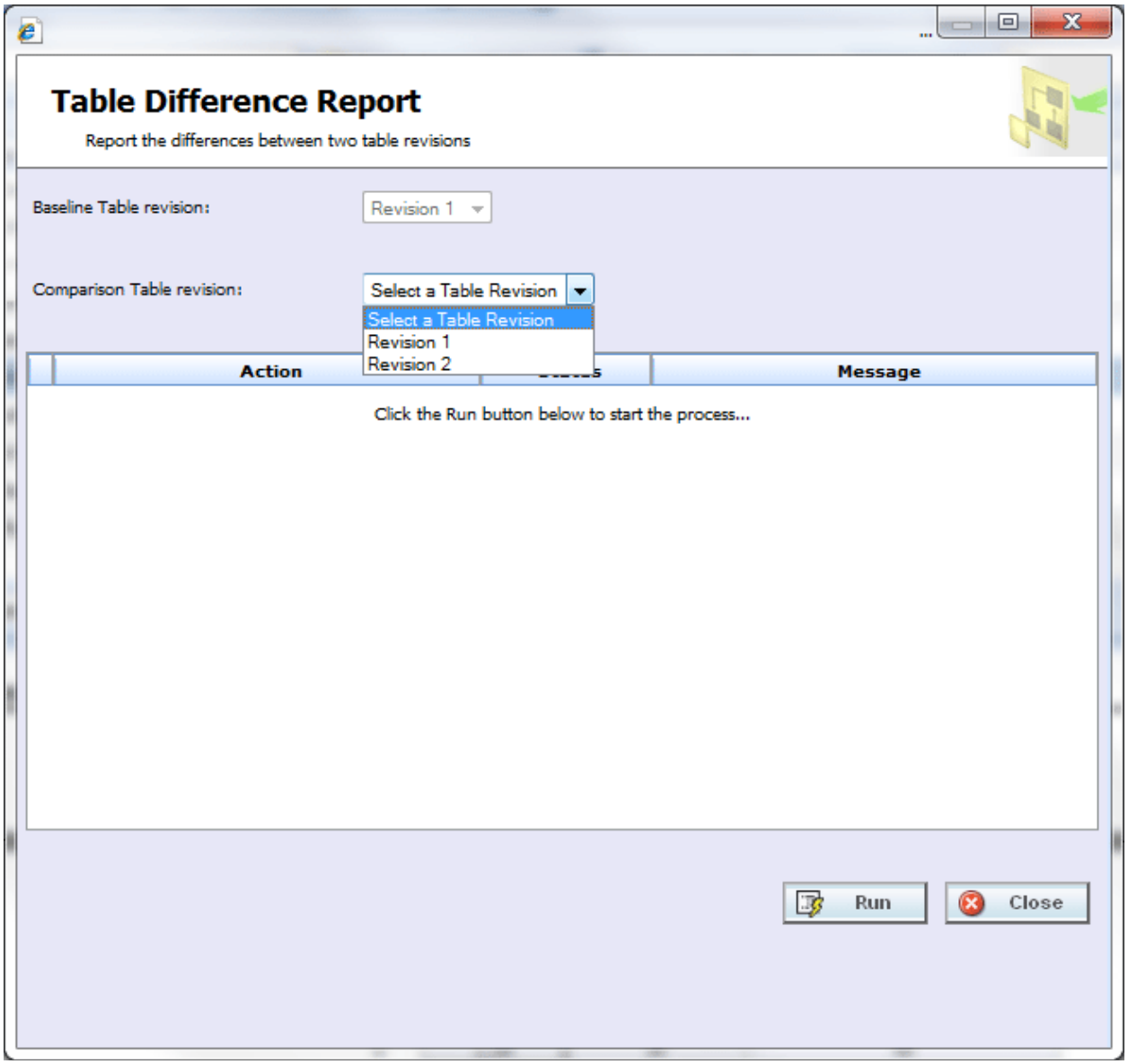


Figure 582 Selecting Revisions to Compare

3. The base revision is determined by where the run report was executed and cannot be changed in the wizard. If this is not the revision you want to serve as base, close the wizard to return to the window that contains that revision you want to serve as base. Run the report from that revision.

4. If the base is correct, select the **Comparison Table Revision**. All revisions are listed. This can be any other revision, either newer or older.

5. Click **Run** to start the report. The progress is displayed as the report runs

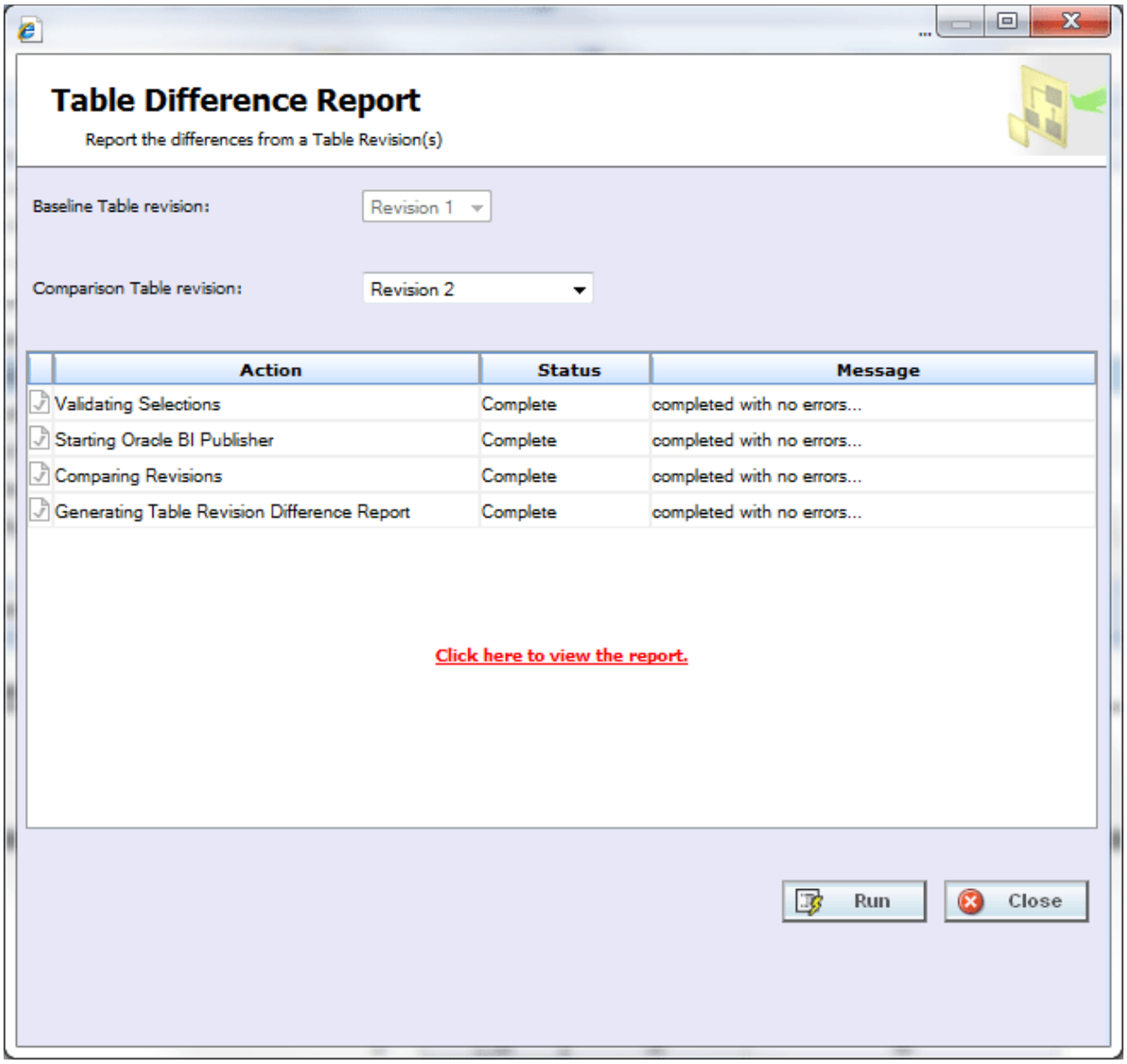


Figure 583 Successful Run of a Table Comparison Report

6. When the wizard finishes, click the "**Click here to view the report**" link. The report is displayed in the Insbridge Document Viewer.

If any errors occurred while running the report or displaying the report, please contact your system administrator.

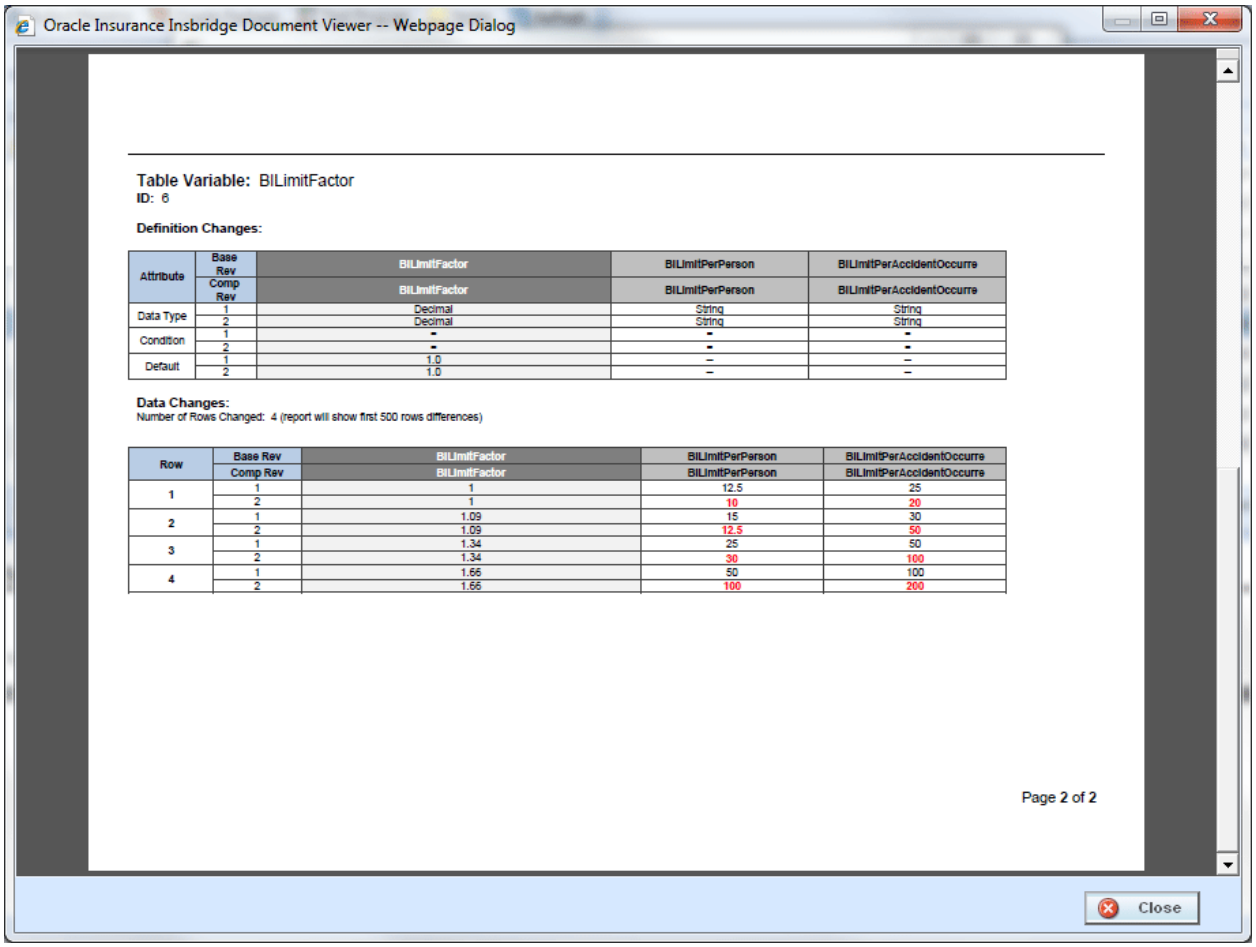


Figure 584 Page 2 of Table Comparison Report

The first page contains a description of the table variable and location of the program from where the report was run. The second page of the table comparison report lists out the details. The definition changes are listed first and data changes are next.

Table Difference Reports are limited to the first 500 rows of data.

Reports are displayed in the Insbridge Document Viewer. These documents are read-only. Reports can be saved to your local hard drive or network or printed. No editing can be done on the Insbridge Document Viewer.

NOTE: *When you do a "Save As" from the Insbridge Document Viewer, you should rename the file. By default, when you go to save a file in the Insbridge Document Viewer, everything is named "InsbridgeDocument". It is recommended that you rename the file.*

VIEW WHERE USED

View Where Used to see if and where a variable or field is used in a program. The View Where Used Report shows all occurrences of the variable or field in other variables, algorithms, or result mappings, in all programs across the project. You also can click through to further see where the variable or field is used in calculated variables or algorithms. The report loads in a window that may be viewed or printed.

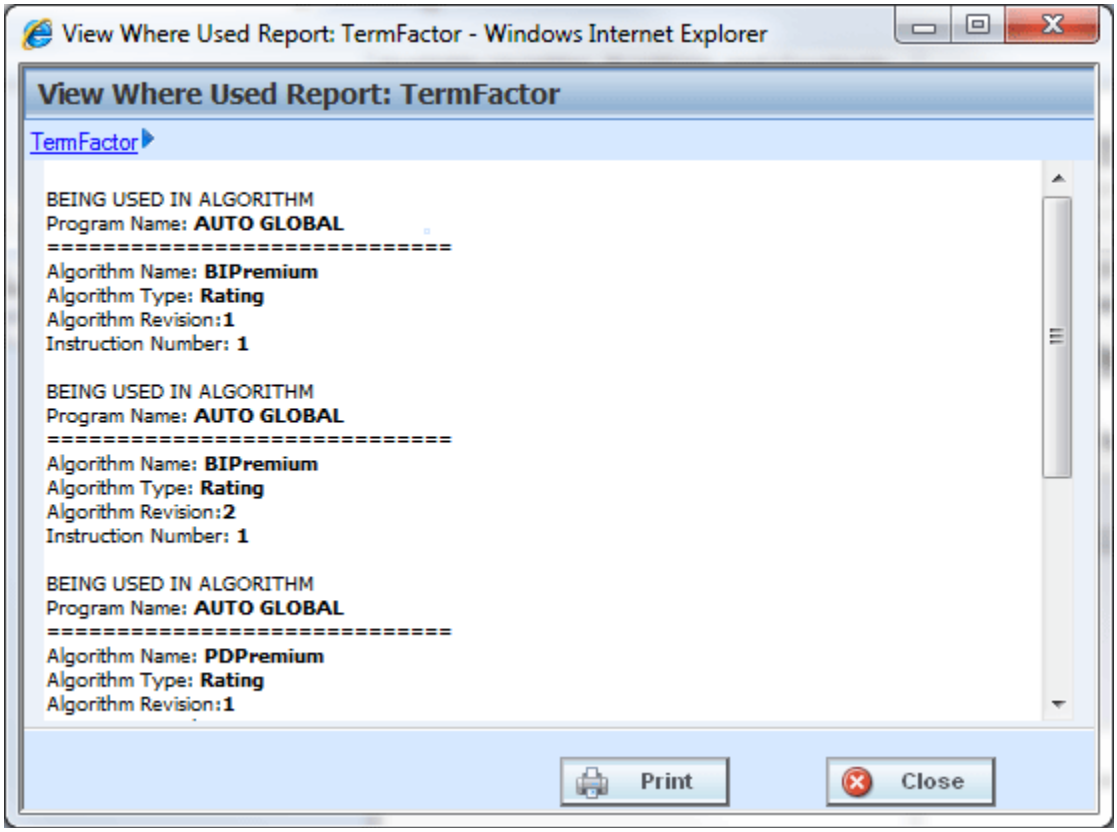


Figure 585 View Where Used Report

If a variable is being used in an Advanced Option, the instruction number will indicate the advanced option that is being used and not the step.

View Where Used Reports are not available on algorithms or driver assignments.

Worksheet usage is displayed in the View Where Used Report. For example, if a calculated variable is included in a worksheet, it will be listed in the Where Used report.

Viewing a View Where Used Report

View Where Used is available when you right click a variable and select **View Where Used** from the menu. Click any hyperlinked variable or algorithm to see the details. The pathway at the top of the screen displays your current location. Click any link in the pathway to return to that item.

Chapter 31

COMMON FUNCTIONALITY

There are some functions that can be performed in many different areas of the Insbridge system such as:

- Locking
- Notes
- Change Control
- Custom Values

LOCKING

Locking can help to protect the integrity of your work and prevent other users from editing elements that you are currently working on. Locking also keeps templates from being altered. This will ensure that the original form of the template remains intact.

There are three types of locks in RateManager:

	Program Version Lock. The program version is locked and all elements contained within this program version are locked. Locked programs or elements are not open for editing. A lock icon indicates that this program or element is locked by a program and is not open for editing or deleting. If the element is locked because the program is locked, you must return to the program and unlock it. This will unlock the element and changes can be made.
	Template Lock. This program version was created from a template. All elements contained within this template version are locked by default. If the element or program is locked because it came from a template, you will not be able to unlock it. You can create a new element or program if necessary.
	In-Use. Elements or programs that are in use are currently being worked on by another user in the system or that you have open in another tab are not open for editing, copying, or deleting. The element or program will be available when the user exits.

Different locks have different properties. A program version lock will close the program version and all associated elements to any changes, additions or deletions. New program revisions can be created, copied, and exported regardless of locking.

A template lock will not allow changes to the element versions that are being used, but new elements and new element versions can be created, changed and deleted.

Only one type of lock icon will be displayed even though more than one kind of lock may be used. A template program can be locked at the program version level. This combination will use the template lock icon with the Version Locked banner.

Notes can be added to any locked element or locked program version at any time.

IN-USE LOCKS

In-Use locks are temporary locks put on an element to indicate that another user is currently editing this element. In-Use locks will lock out other users while you are editing the element. This prevents other users from making changes that you may not be aware of or that do not need to be made.

An element is locked automatically when you enter the edit tab. The element is unlocked when you exit the edit tab, not when you save.

An element displays the lock icon of the active revision. For example, if you have a variable with two revisions and revision 2 is the active revision. When you edit the active revision: revision 2, the red element lock icon is displayed by the variable name and the revision you are editing. If you edit the non-active revision, a lock icon is displayed next to the revision you are editing and not next to the variable name.

Single elements such as categories, input variables, result variables, constants, program sequencing, and result mapping are closed to editing when a lock is in place. Revisioned elements, such as table variables, calculated variables and algorithms have the selected revision closed to editing when a lock is in place. This means that if revision 1 is locked, revision 2 may be open for editing. If only one revision exists, the element is locked from any further editing, deletion or copying.

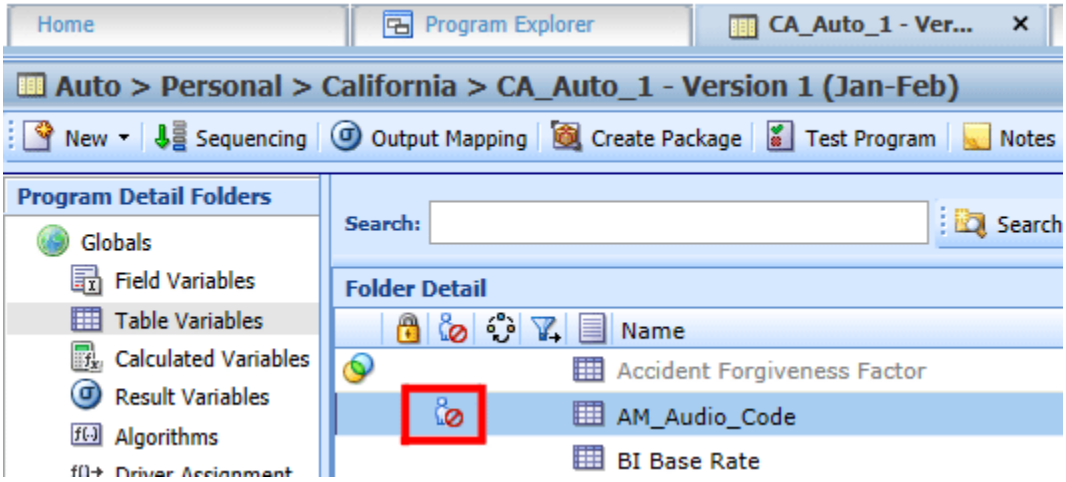


Figure 586 In-Use Locking on an Element

Any open element can have an element lock placed on it when a user goes to edit it. An element lock will be placed on the element you are currently editing. If you are editing an algorithm and open up a new tab to look at inputs, the algorithm you are editing will display an In-Use lock until you close the editing tab.

Elements that are locked by a program lock or a template lock are not open for editing.

Releasing an In-Use Locked Element

Hover your cursor over the In-Use lock icon to identify the user who is editing this element and how long they have been editing it.

If an element has been locked for an extended period of time, please contact your system administrator. The system administrator can release the element from the user who is editing by ending their session.

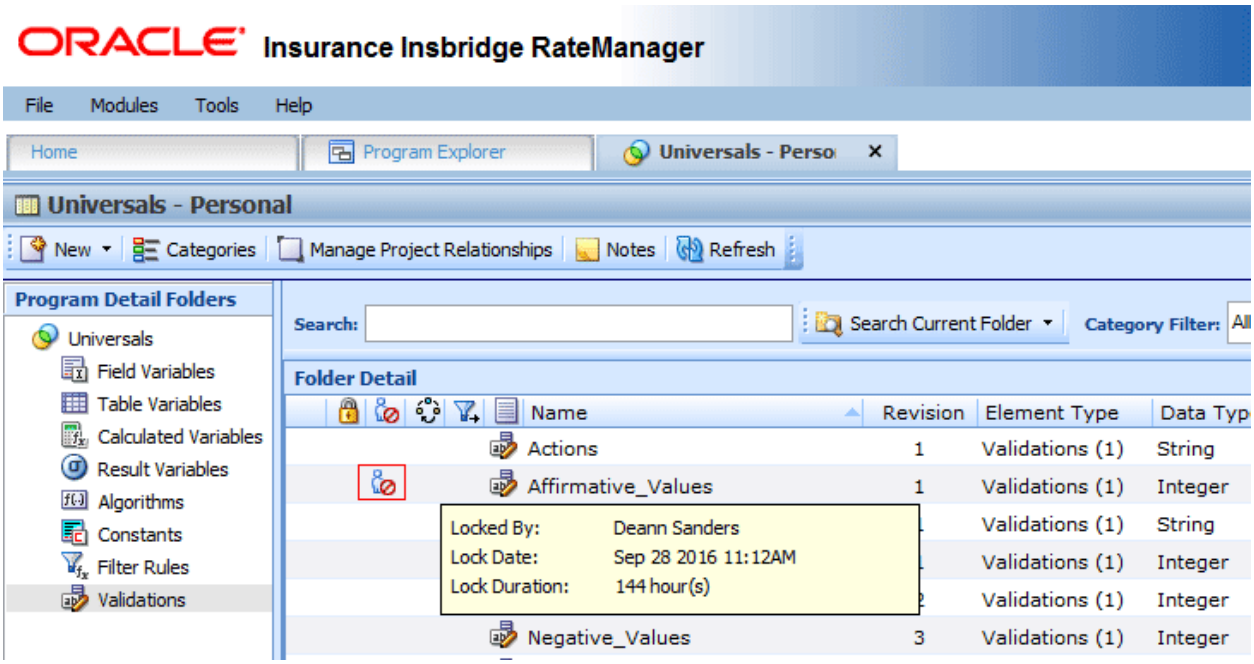


Figure 587 In-Use Message

If you attempt to open a locked element, a message will be displayed. The message lets you know who is editing the element and how long the element has been locked.

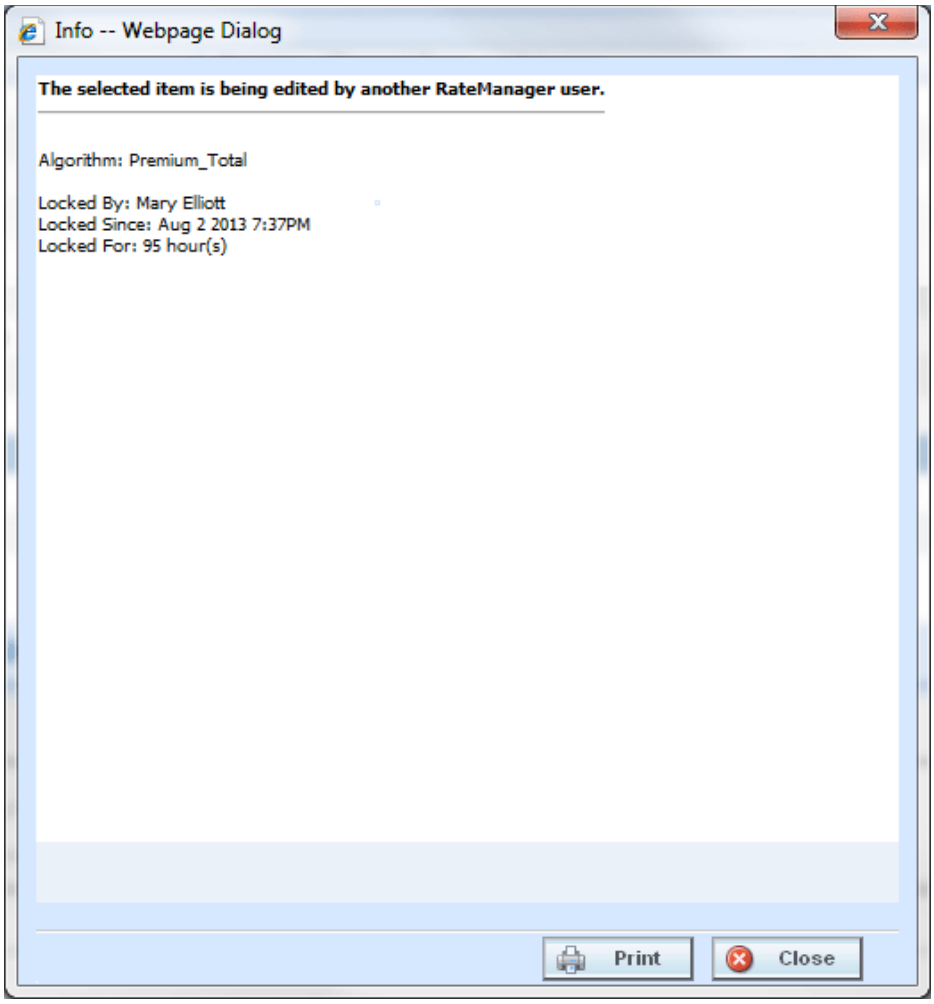


Figure 588 In-Use Lock Information

PROGRAM VERSION LOCKS

An administrator can lock a program version to prevent it from being changed or deleted. This locks all the selected versions of the Variables, Algorithms, and Driver Assignments elements plus the Sequencing and Result Mappings used in this program version. The element versions are locked throughout the project based on the last packaged date. Changes to these element versions in this project are not allowed, regardless of what folder or program version you are working in. Locking a version does not affect other projects.

For example, you have an auto program with three versions. Program version 1 uses version 1 variables, algorithms and driver assignments. Program version 2 uses version 2 variables, algorithms and driver assignments. Program version 3 uses version 3 variables, algorithms and driver assignments. If you lock program version 2, all version 2 variables, algorithms and driver assignments are locked in all three program versions. You cannot edit or delete any of the local elements versions in the program or the global element versions in the product. Universal elements remain locked for any program in the project. Elements in program versions outside of the version that have the lock applied are open for copying. Elements that are locked in version 2 are not open for copying. The same elements in version 3 are open for copying.

On The Program Details tab there is a program lock icon indicating which elements are locked at the program version level. Locking a program version also locks the associated Categories, Fields, Variables, Algorithms, and Driver Assignments throughout the product. Lock icons are not displayed on the Program Explorer tab.

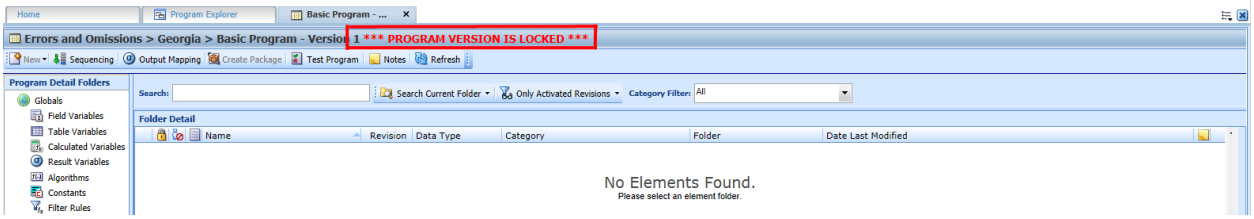


Figure 589 Locked Program Version Banner

When you enter a program version that is locked, a **VERSION LOCKED** banner is at the top of the tab. Other program versions of that program do not have the banner but will have the lock icons on shared elements indicating the elements that are locked by the locked program version.

Locking a Program Version

1. Navigate to the **Program Explorer** tab. Select the , folder, and program that contains the program version you want to lock.

2. Place a check in the **LOCK** check box for the version you want to lock.

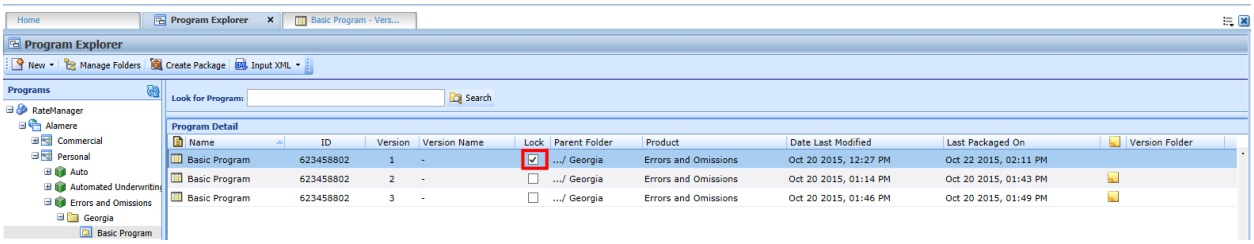


Figure 590 Selecting a Program Version to Lock

3. A message is displayed. Click **OK**. This version is now locked.

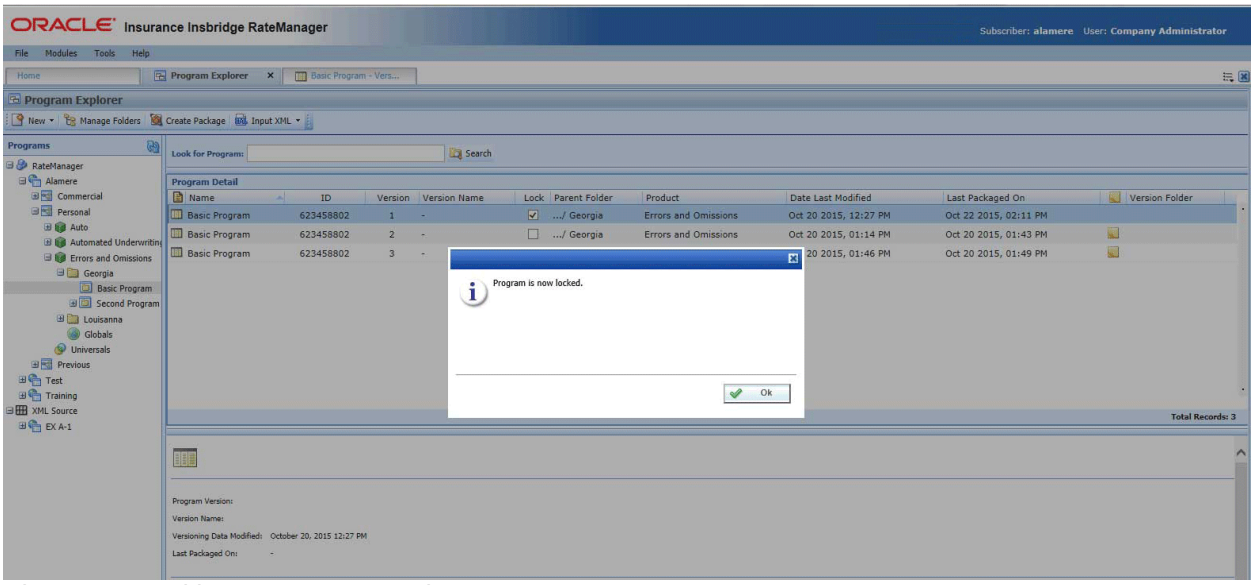


Figure 591 Locking a Program Version

You will not be able to make any changes to the version or to any associated items while it is locked. If you try to edit, a message will be displayed telling you the item is locked.

Only program versions that have been packaged can be locked. If you have made changes to the program or if you packaged the program a while ago, you should package before locking.

Your screen and right click menu options will change when a program lock is put in place.

- The New, Delete, and Create Package options along with the Copy, Show/Hide Revisions and Delete options will be removed from the right click menu of the program elements.
- Result groups will have the New Group and Copy Group options removed. Edit Groups will have the Save option removed.
- Program Sequencing will have the Save Sequencing and Change Sequence Order removed.
- Notes can still be added to a locked program version.

Unlocking a Program Version

Unlocking a program version frees up all items for editing and deletion. You must unlock a program before you can package it or reconcile.

1. Navigate to the **Program Explorer** tab. Select the , folder, and program that contains the program version you want to unlock.

2. Remove the check in the **LOCK** check box for the version you want to unlock.

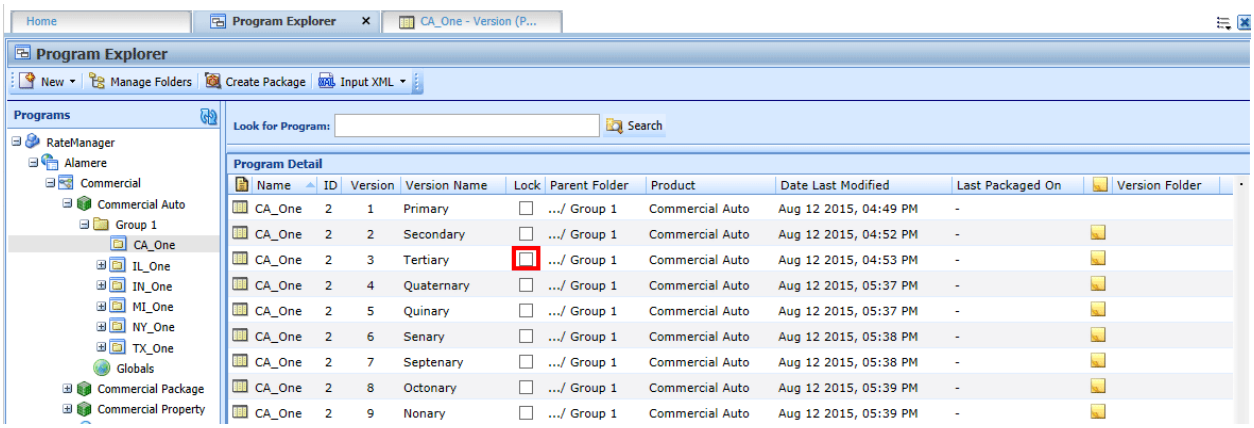


Figure 592 Selecting a Program Version to Unlock

3. A message will be displayed. Click **OK**. This version is now unlocked.

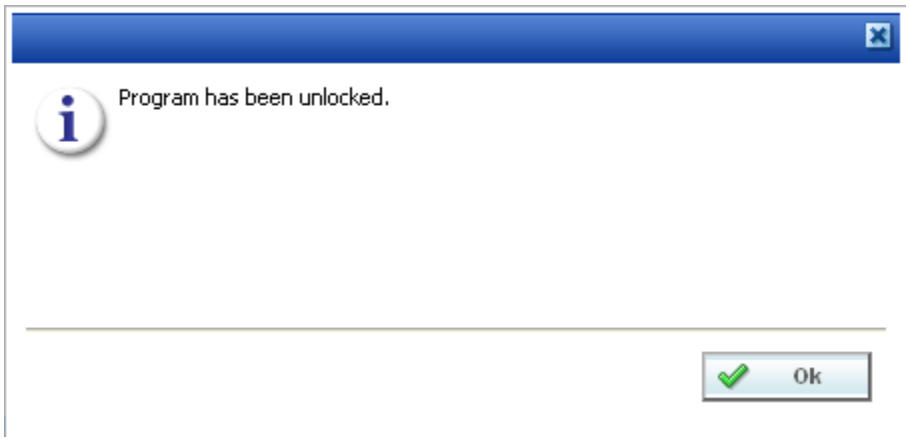


Figure 593 Unlocking a Program Version

TEMPLATE LOCKS

Template locks are automatically applied to any template brought into RateManager from the Library. A template lock locks all elements of the program from any changes. A template lock does not prevent the program from being packaged or reconciled. A template lock cannot be removed.

Single elements such as categories, fields, result variables, program sequencing, and result mapping are closed to editing or deletion by a template lock. Revisioned elements, such as table variables, calculated variables and algorithms have the selected revision closed to editing or deletion by a template lock. There is a template lock icon, indicating which revisions are locked. If the active revision is locked by a template lock, the element name will have a template lock icon also.

Even if the active revision changes, the template locked revision cannot be edited or deleted. You can add new elements. The new elements do not have a template lock. You can edit or delete as needed.

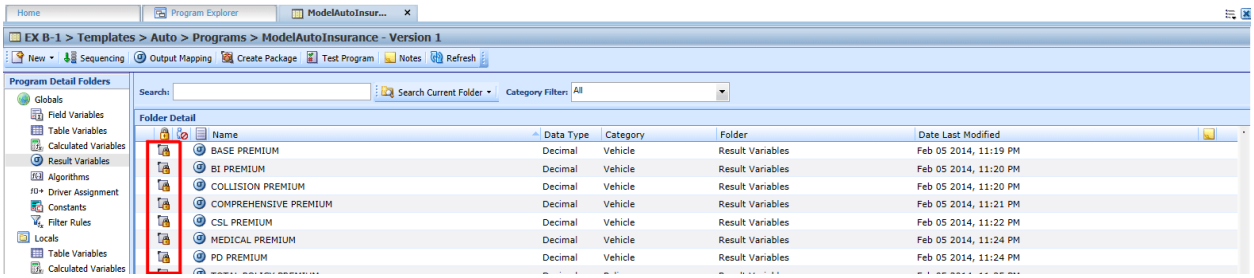


Figure 594 Template Locks

Notes can be added to a template locked element at any time.

NOTES

Notes allow for explanations or comments to be added to an element or a program version. These optional entries can be added to program versions, variable revisions, algorithm revisions, sequencing, result mapping, categories, and fields. Notes can be added from many screens throughout RateManager.

From the main notes screen, you can add, edit, or delete a note. You can enter as many notes as you need at any time you want. On the active elements of a program locked version, notes can be added, edited or deleted.

When an element or program version has a note, the notes icon is displayed towards the end of the program version or element. The notes cannot be placed at the program level, only at the version level. The notes icon is displayed at the element level, but only if the active revision has a note. If only a non-active revision of an element contains a note, the notes icon will not be displayed at the element level.

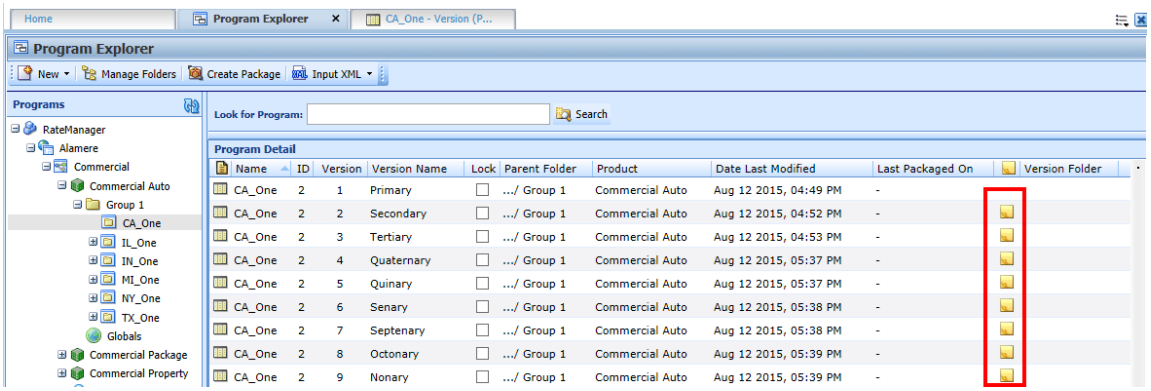


Figure 595 Notes on a Program Version

To view the note, highlight the element and click NOTES. Notes are displayed in a separate screen.

- Notes are entered on all linked table variables. For example, if you have a table variable that is linked to 10 other table variables and you add a note, then the note is placed on the original variable and on the 10 table variables that are linked to it.
- A note entered on a universal variable can be viewed on the universal tab and the program details tab. A note entered on a global variable can be viewed on the globals tab or the program details tab.
- Program versions must be selected for a note. You cannot enter a note at the program level. You must select the version where you want the note placed.
- Elements do not require a specific revision to be selected. Elements default to the active revision. If you select a variable, but not a specific revision, the note is placed on the active revision. If you want to place a note on a specific revision, you must select the revision and add the note there.
- Notes entered on revision one are not displayed on revision 3 of the same element. If the same information needs to be on all revisions, you will have to enter a note on all revisions.
- Copying a program version or creating a new version automatically creates a note on the new version with details regarding where the new program version was copied or created from.
- Notes are not carried through to copied programs or elements. Only the original program version or element will contain the notes.

NOTE: *Notes are not carried through to packages.*

Adding Notes

Notes can be added at any time. A note can contain up to 200 characters including spaces. Certain special characters are restricted:

Name	Character
Ampersand	&
Asterisk	*
Backslash	\
Greater-Than Sign	>
Less-Than Sign	<
Question Mark	?
Quotation Mark	"

All other characters are allowed.

When you enter your note, you can type a note in directly or you can paste a note in from an outside source. Be aware that you may type or paste in more than 200 characters, but the note only saves the first two hundred. Any extra characters are cut off. You cannot save a blank note.

1. Click NOTES and the main notes screen is displayed.

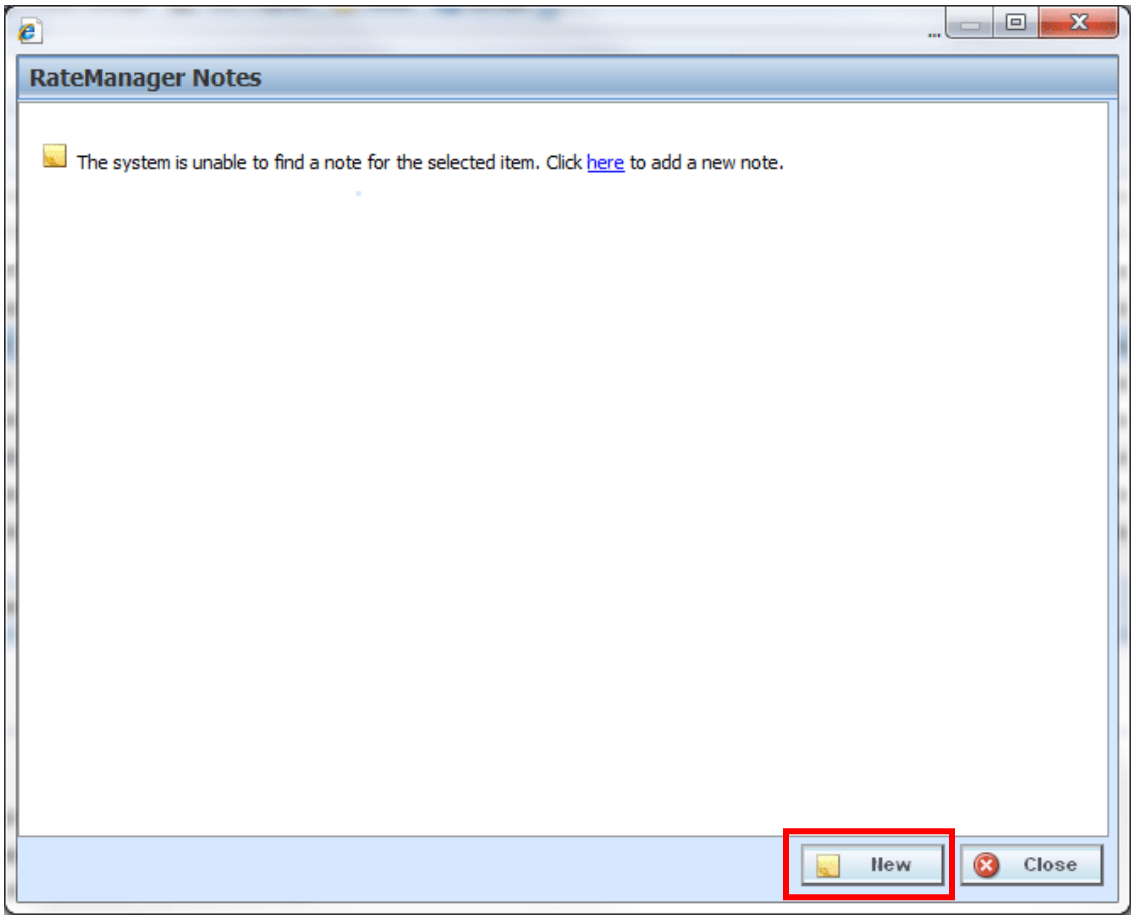


Figure 596 Adding a Note to the Main Notes Screen

2. The *Main Notes Screen* holds all the notes and change controls entries that have been made on this element or program version. Notes are listed in date order of newest to oldest with the newest note first. If no notes have been entered, the screen will have the option to enter a new note. You can click the hyperlinked here or click **New**. The notes details screen is displayed.

3. The *Notes Detail Screen* contains the details of the note and/or any attachments. Enter your **Note** and/or attachment.

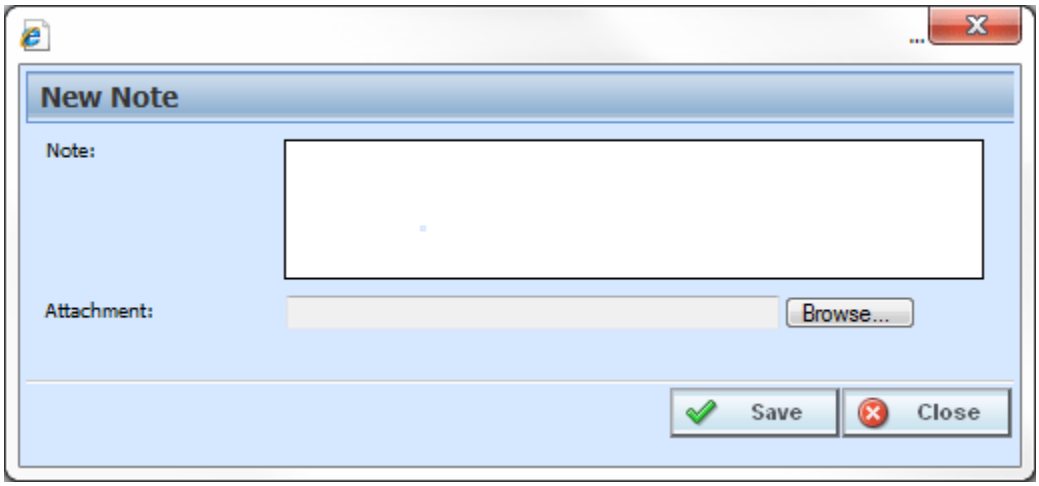


Figure 597 Completing the Notes Detail Screen

4. Click **Save**. The notes detail screen will close and you will be returned to the main notes screen. Your note will be listed.

NOTE: *You must click Save if you want to save the note. Clicking **Close** on both the Main Notes screen and the Notes Detail screen will close the screen without saving the note.*

5. Enter any additional notes, if needed.

6. When you are finished, click **Close** to close the screen.

SCREEN OPTIONS

A blank Notes screen has two buttons, New and Close. A populated Notes screen also allows for editing and deleting. The Notes Detail screen has Save and Close.

	New: Opens a Notes Detail screen where you can enter and save your note.
	Close: Closes the screen. If you have not saved your note entry, this closes the screen without saving any additions or changes.
	Save: Saves your entry.
	Edit: Opens the selected entry for editing. The notes are saved under the new date and time. Change Controls cannot be edited.
	Delete: Removes the selected entry.

Attaching Files

You can place attachments on a note. Attachments can be text format, MS Word documents, MS Excel spread sheets or Adobe Acrobat PDF files. Other files types will result in an error message being displayed. File names are limited to 25 characters. If your file name has more than that, any extra characters are cut off.

NOTE: *A note can have only one attachment. If you have to attach more than one file, you will have to create a new note for each attachment. Three attachments requires three separate notes. The notes can all be on the same element or program.*

1. To attach a file, on the notes detail screen, click **BROWSE…**, this pulls up your computers dialog box.

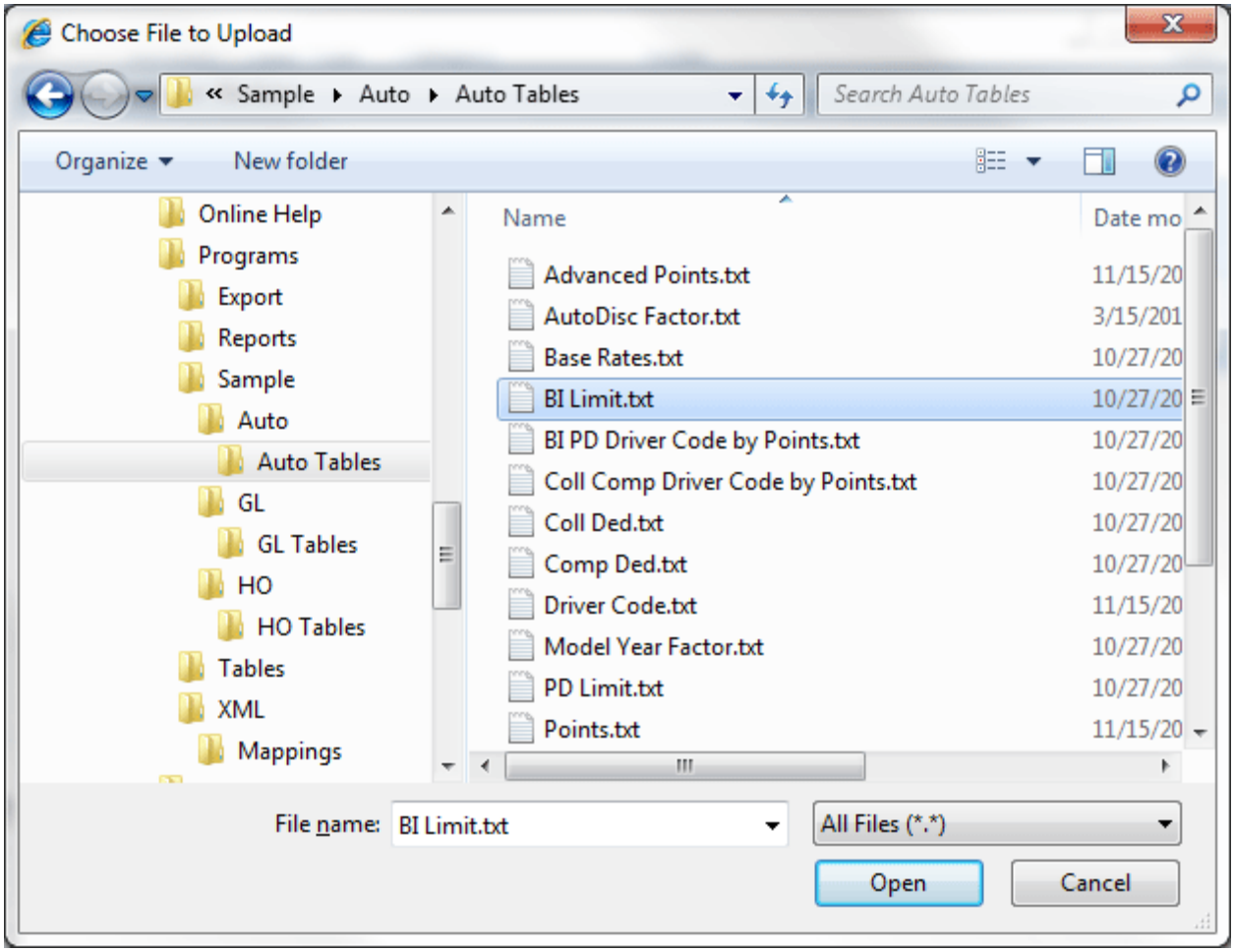


Figure 598 Dialogue Box

2. Find your file and click **Open**. This closes the dialog box and returns you to the notes detail screen.

3. Your file is displayed in the Attachment field. Attachment file size is limited to 10MG per file. Each note holds one attachment. There is no limit on how many files you place in the notes area.

4. Attachments must have an entry in the **Note** area.

5. When you are finished, click **Save**. The notes detail screen closes and you will be returned to the main notes screen. Your note and attachment are listed.

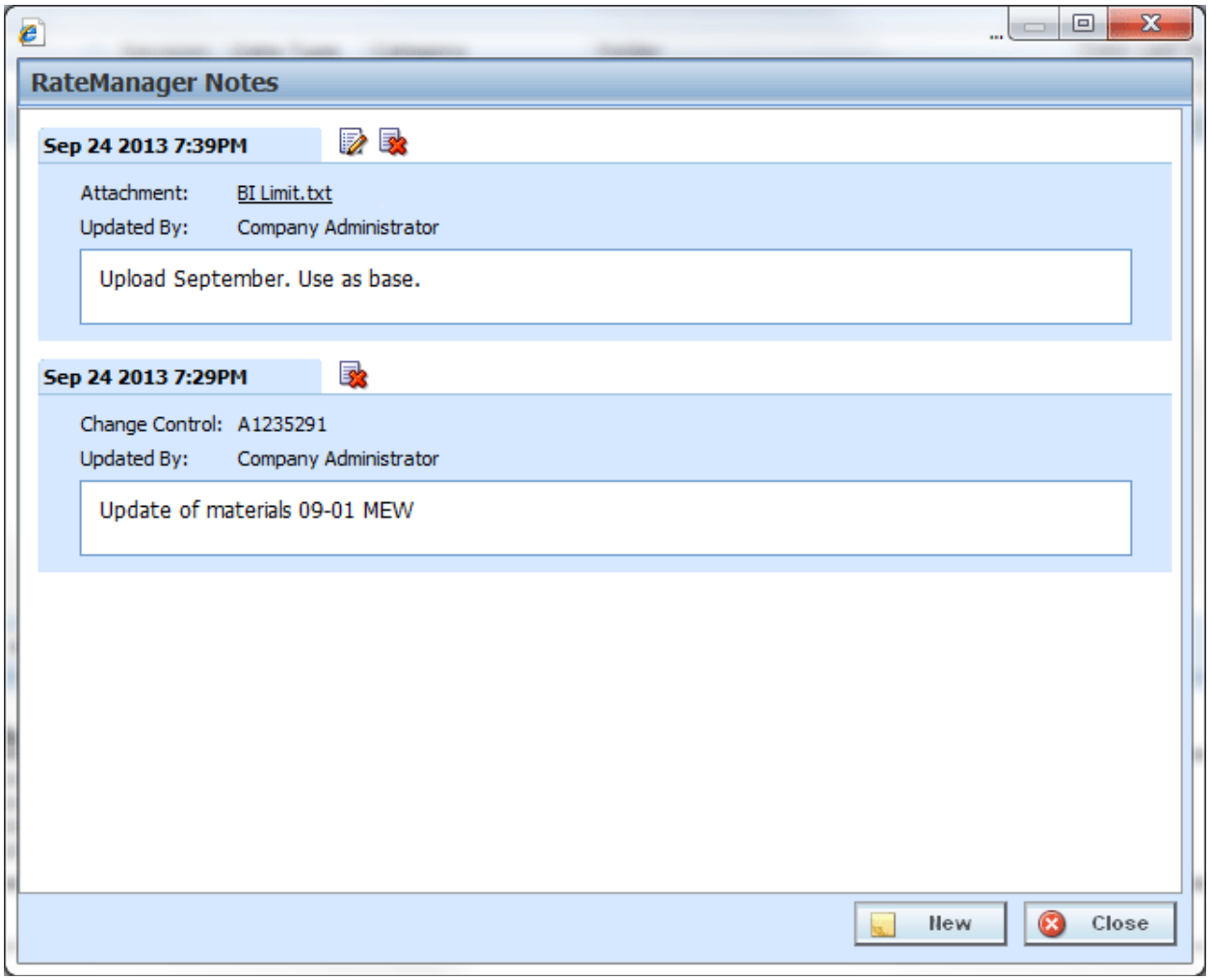


Figure 599 Completed Notes Screen

6. Enter any additional notes, if needed.

7. When you are finished, click **Close** to close the screen.

NOTE: *Certain file types will be denied, such as exe, com, vbs, or dll. If you try to attach a document with this file extension, the command will be denied.*

Viewing Attachment Files

After you have saved the note, you can view the attachment. Click the attachment file you want to view. Your attachment is displayed in a separate window, depending upon file type. Word documents are displayed in MS Word, Excel documents are displayed in MS Excel, and PDF documents are displayed in the Insbridge document viewer. These documents are read-only. You may be able to do a “Save-As” copy to your local hard drive but any changes you make are not made to the attachment on the note.

If the attachment needs to be edited, you need to go to the original source and edit there. For example, if you have uploaded a Word document, you need to open the original file in Word, make your changes there and save the file. Then you can return to your note and reattach the updated file.

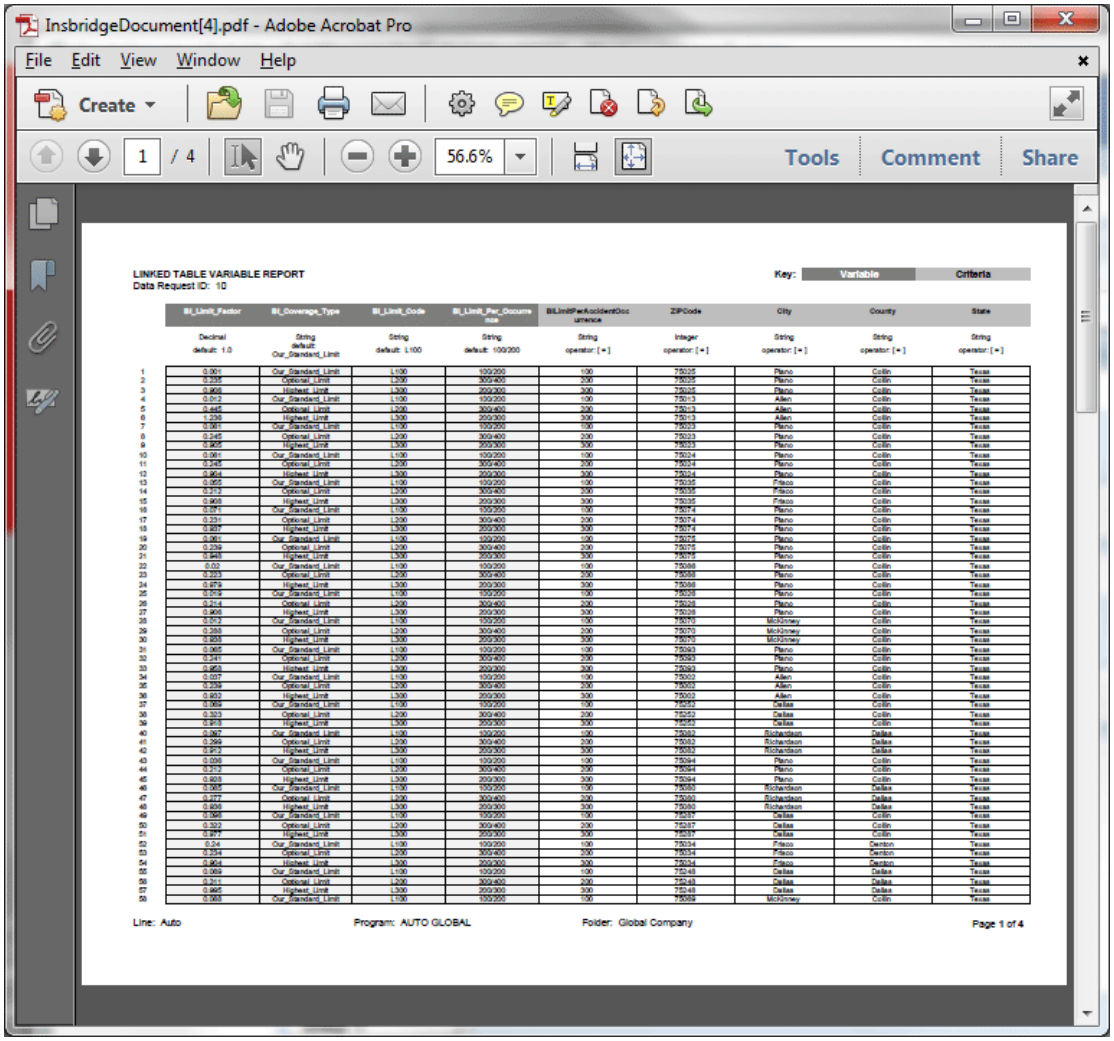


Figure 600 PDF File in the Insbridge Document Viewer

NOTE: *When you do a “Save As” from the Insbridge Document Viewer, you should rename the file. By default, when you go to save a file in the Insbridge Document Viewer, everything is named “InsbridgeDocument”. It is recommended that you rename the file.*

Remove Attached Files

Attachments cannot be deleted or edited. To remove an attachment, you can delete the entire note. If you would like to replace the attachment, you can edit the note and select another file. This replaces the first attachment.

Editing Notes

Notes can be edited at any time. You cannot replace or edit attachments. If you must update or replace an attachment, you have to delete the previous Note and create a new Note with the updated attachment.

1. Highlight the program version or element that contains the note you want to edit.
2. Click **NOTES** and the main notes screen is displayed.
3. Click **Edit** on the note you want to edit. The notes detail screen for that note is displayed.

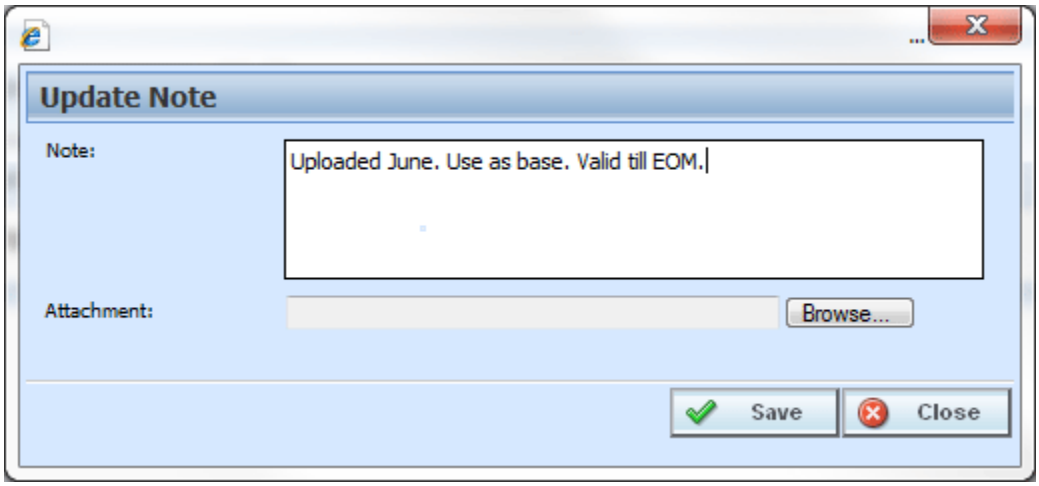


Figure 601 Editing a Note

4. Make your changes.
5. Click **Save** to save your entry and return to the main notes screen.
6. When you are finished, click **Close** to close the screen.

Deleting Notes

Notes can be deleted at any time. If you delete the last note, the notes icon is not displayed on the program version or element. Deleting a Note deletes any attachments on the Note.

1. Highlight the program version or element that contains the note you want to delete.
2. Click **Notes** and the main notes screen opens.
3. Click **Delete** on the note you want to delete.
4. A warning message is displayed. Click **OK** if you want to delete the note. Cancel returns you to the notes detail screen without any changes being made.

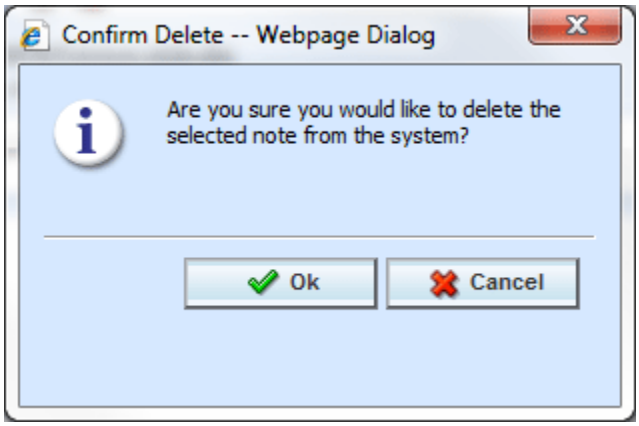


Figure 602 Deleting a Note

5. Click **Close** to close the screen.

NOTE: *Please take care when deleting notes. This action cannot be undone.*

CHANGE CONTROL

Change control allows for a tracking identifier and justification to be added to an element revision or a program revision. If change control has been activated, it is mandatory to make an entry to any element or program when it is revisioned. If change control is not activated, the option is not presented. Any program or program element that can be revisioned can require a change control entry. Change control will only be required at the time the program or element is revisioned.

- Change control uses the same screen as the notes feature. From the main notes screen, you can delete a change control. No additional change controls can be added.
- Change controls cannot be edited.
- Only one change control per revision is allowed.
- When an element revision or program version has a change control, the notes icon is displayed at the end of the program version listing or element revision listing. The notes icon is not placed at the program level, only at the version level. The notes icon is displayed at the element level, but only if the active revision has the change control. If only a non-active revision of an element contains a change control, the notes icon is not displayed at the element level.
- Change controls are entered on all linked table variables. For example, if you have a table variable that is linked to 10 other table variables and you create a new revision, then the change control is placed on the new revision of the original variable and on the new revisions of the 10 table variables that are linked to it.
- A change control entered on a universal variable can be viewed on the universal tab and the program details tab. A change control entered on a global variable can be viewed on the globals tab or the program details tab.
- To view the change control, highlight the element and click Notes. The change control note is displayed in a separate screen.

NOTE: *Change controls are not carried through to packages or copies.*

Activating Change Control

System Administrators can activate Change Control on the Tools→Preferences→Administrator Web Settings. Select ON to activate Change Control. If change control is on, it will be mandatory for users to enter a change control entry and justification to any element or program when it is revisioned. If change control is off, the option will not be presented.

Adding Change Control

A change control is added at the time the revision is created if change control is required. The change control option must be turned on by an administrator. When activated, change control entries are mandatory. If the option is deactivated, no change control option will be presented.

A change control justification can contain up to 200 characters including spaces. Certain special characters are restricted:

Change Control restricts:

Name	Character
Ampersand	&
Asterisk	*
Backslash	\
Greater-Than Sign	>
Less-Than Sign	<
Question Mark	?
Quotation Mark	"

All other characters are allowed.

When you enter your change control, you will be able to type the justification directly in or you can paste a justification from an outside source. Be aware that you may type or paste in more than 200 characters, but the justification field will only save the first two hundred. Any extra characters will be cut off. You must make an entry. Blank change controls are not allowed.

1. Click the element or program where you want to create a new revision.

2. If change control is required, a separate screen opens.

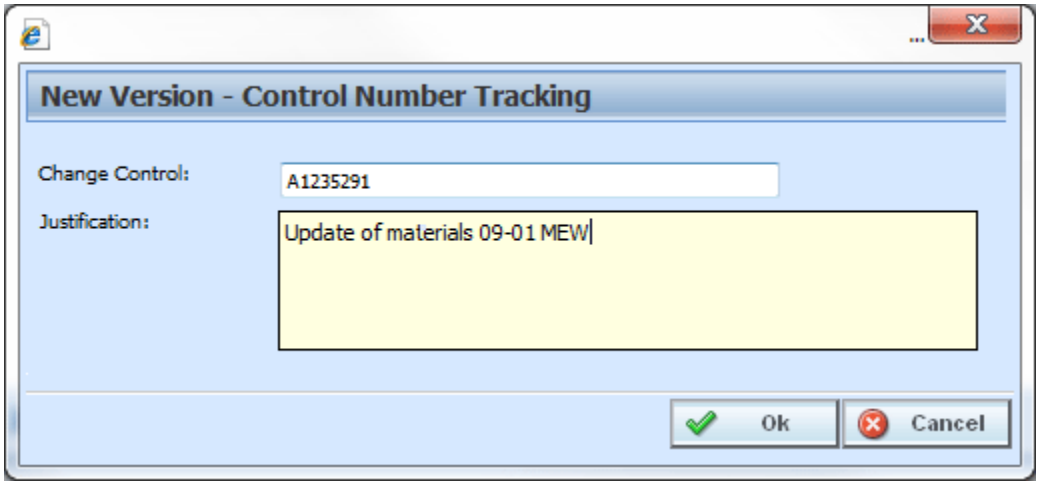


Figure 603 Change Control

3. Enter the change **control ID** and the **justification**.

NOTE: *Table Variables can select to include data in the new revision.*

4. Click **OK**. The change control screen closes and you are placed on the revision details screen for the element or program version.

5. When you complete your revision, your change control is listed on the notes screen for that revision. Notes may be added after the revision has been created. Notes are listed in date order of newest to oldest with the newest note first.

NOTE: *Verify your entry before saving. Change controls cannot be edited.*

Deleting Change Control

Change controls can be deleted at any time. If the change control is the only item on the notes screen and you delete it, the notes icon will not be displayed in front of the program version or element.

1. Highlight the program revision or element revision that contains the change control you want to delete.
2. Click **Notes** and the main notes screen is displayed.
3. Click **Delete** on the change control you want to delete.
4. A warning message is displayed. Click **OK** if you want to delete the change control. Cancel returns you to the notes detail screen without any changes being made.

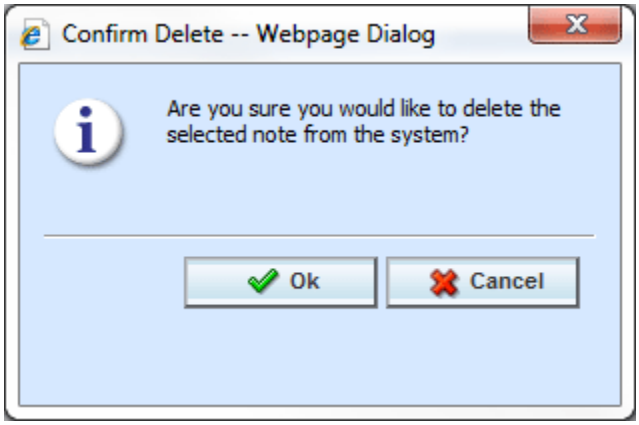


Figure 604 Deleting a Change Control

5. Click **Close** to close the screen.

NOTE: *Please take care when deleting change controls. This action cannot be undone.*

CUSTOM VALUES

Custom values can be used in table variable masks, calculated variables, algorithms, drivers assignments, and filter rules.

A custom value is a fixed value that you created. For example, if you wanted to use .5 in an arithmetic step, you could create a custom value for that purpose. The custom value created is for that step only. It is not displayed any place else. If you want to use this value again, you will have to enter it in again. If this is a value that you use frequently, you may want to create a user defined constant with that value.

Adding a New Custom Value

A new custom value can be created for a table variable mask or while entering or editing a calculated variable, algorithm, drivers assignment, or filter rules without leaving the edit tab.

Custom values can be created from:

- Right click menu on an algorithm step input.
- Adding a mask to a table variable.
- Selecting **Custom Value** from the System Level Constants menu on any calculated variable, algorithm, drivers assignment, or filter rule. Scroll down to the bottom of the system level constants menu options. Select **Custom Value**.

To Add a Custom Value

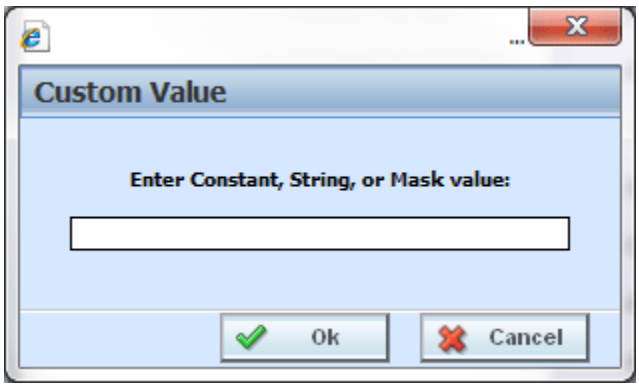


Figure 605 Custom Value

1. Enter in a **Value**.
2. Click **Ok** to save your entry. The window closes and your new custom value is populated in the input.

CONTACTING SUPPORT

If you need assistance with an Oracle Insurance Insbridge Enterprise Rating System product, please log a Service Request using My Oracle Support at https://support.oracle.com/.

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Address any additional inquiries to:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

INTERNET EXPLORER SETTINGS

RateManager is a web application that requires certain permissions and controls in order to properly interact with your desktop and file system. You may need to change or adjust your Internet Explorer settings to fully utilize all of RateManagers features.

HOW SECURITY LEVELS AFFECT RATEMANAGER

Different security levels affect how RateManager functions. The basic security levels **(High, Medium, Medium-Low** and **Low)** affect RateManager in slightly different ways.

High	RateManager will not function properly when security is set to **High.** Security must be **Medium** or lower.
Medium	Only files located on a network share can be batch rated, i.e. the file path must be of the form "\\server\folder\file.xml". Files located on your machine or on a mapped network drive may not be available. When batching a file, you may be prompted with a warning message to get the file from the server. You must click **OK** in order to rate the file.
Medium-Low	RateManager functions the same as with the security setting of **Medium**.
Low	Not Recommended.

Fakepath

Internet Explorer uses HTML5, which contains a file upload control that should not reveal the real local path to the file you have selected if you manipulate its value string with JavaScript. Instead, the string that is returned by the script, which handles the file information, is c:\fakepath.

In IE, the real path to the file will be shown only if the page that contains the control is added to the trusted sites collection of the browser.

This means that when using IE, RateManager users who try to rate a file in the Testing Module, the Releases Module and the Impact Analysis Module may see the file path as C:\fakepath in the XML Input section of the page.

This may be resolved by adding Insbridge as a Trusted Site.

Adding a Trusted Site

1. On IE, select Tools→Internet Options→Security.
2. Select **Trusted Sites**.

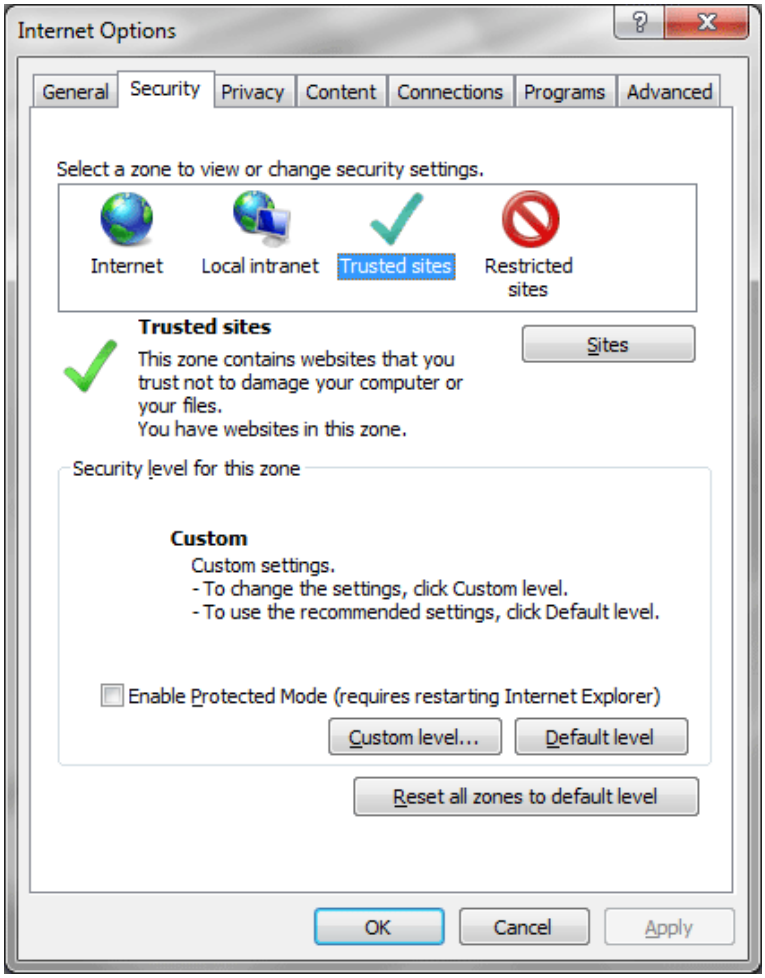


Figure 606 Selecting Trusted Sites

3. Click **Sites**. A separate window is displayed.

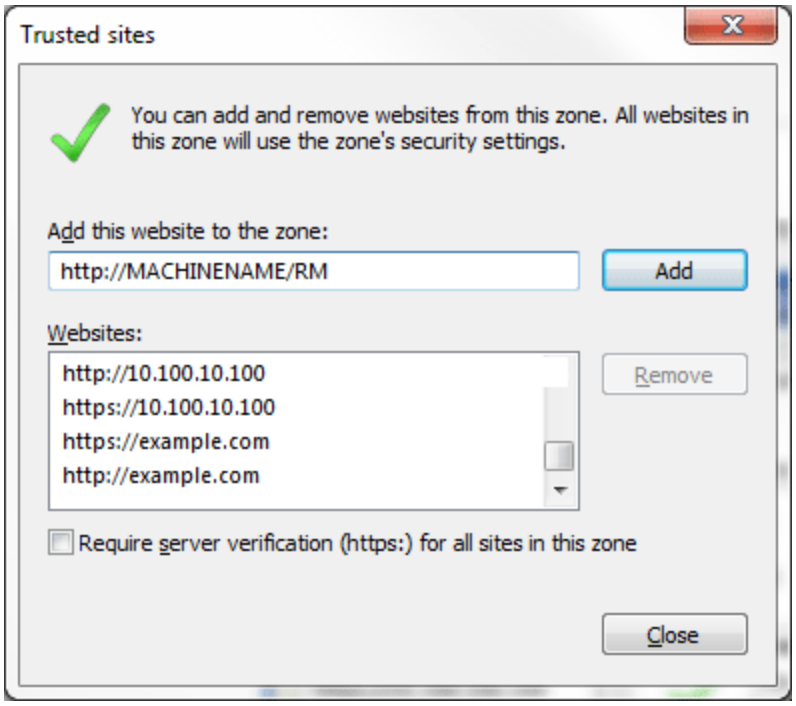


Figure 607 Adding RateManager to Trusted Sites

4. Add your Insbridge site as a trusted site. You may need to uncheck the Require server verification (https:) for all sites in this zone. If you do not know your Insbridge site, ask your system administrator.

5. Click **Close**.

6. Click **OK** on Internet Options to save your selections and close the window.

This allows auxiliary screens from Insbridge to open.

Removing the URL Bar on RateManager

If the URL is displaying in your screen and you would like to remove it, you may want to add your Insbridge site as trusted site and then enable the option to allow script initiated windows.

1. On IE, select Tools→Internet Options→Security.

2. Select **Trusted Sites**.

3. Click **Sites**. A separate screen is displayed.

4. Add your Insbridge site as a trusted site. You may need to uncheck the Require server verification (https:) for all sites in this zone. If you do not know your Insbridge site, ask your system administrator.

5. Click **Close**.

6. Back on the Security tab, select **Custom Level**. This opens a separate screen.

7. Scroll down to the Miscellaneous section.

8. **Enable** the option to *Allow script-initiated windows without size or position constraints*.

9. Click **OK** to save your changes here and then Click **OK** to save all your changes.

Failure to Display Popups

Popups are screens that automatically open when you click a web site or a hyperlink. Popups frequently are annoying ads and many users do not want popups appearing on their system so they install or activate popup blockers. Popup blocker programs will severely limit any type of auxiliary screen from being displayed. Insbridge contains many information screens that are critical for users to view but are considered auxiliary screens by popup blocker programs.

You may not even be aware you have a popup blocker installed because popup blockers are included in newer versions of Windows many popular browser toolbars and other Internet service software.

NOTE: *If at any time you are unsure how to enable popups, please contact your system administrator.*

You can disable popups to allow for select content to be displayed.

Microsoft IE 11

Turning off Pop-up Blocker:

1. Open an Internet Explorer window.
2. Click Tools.
3. Locate Pop-up Blocker on the menu. The options are on or off. If you want to leave pop-up blocker on, add RateManager to the exceptions list. The settings menu is available only when Pop-up Blocker in on.

Pop-up Blocker is also available in the Tools→Internet Options menu:

1. Click the Privacy tab.
2. Uncheck Pop-up blocker.
3. Click OK to save your changes.

This allows popups to be displayed.

NOTE: *You also can check your failed popups option and select to add Insbridge.*

If this fails to produce the desired results, contact your system administrator for further assistance.

If Pop-up Blocker is on, you may get an information bar displayed at the bottom of the screen informing you that a popup window was blocked. You can allow popup from RateManager by selecting to always allow from this location.

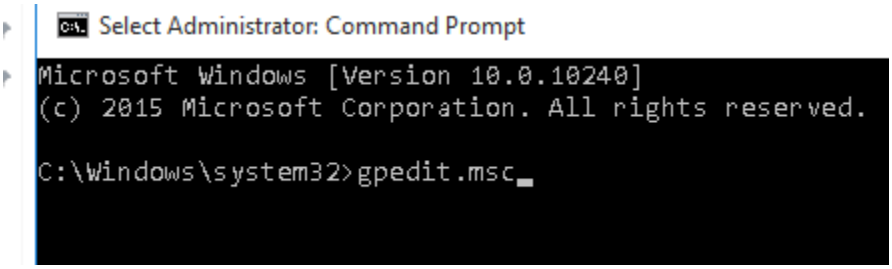
Microsoft Edge

RateManager requires IE 11. In Microsoft Windows 10, Microsoft Edge is the default browser. RateManager does not open in Edge but can be opened on a Windows 10 machine by starting IE from the Go/Start menu and then opening RateManager. There is an option in Edge that allows for intranet traffic to automatically be opened in IE. This will apply to all intranet traffic, not just RateManager. This is an optional step.

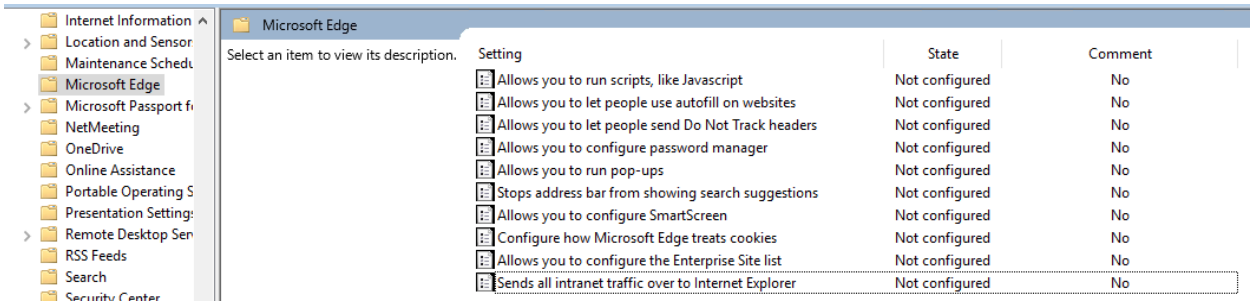
By turning this feature on, and adding the RateManager server to the intranet sites, RateManager will automatically be opened in IE in a Windows 10 machine.

Steps to Invoke IE11:

1. Add the Insbridge URL to your Local Intranet sites.
2. Open a Command Prompt (as Administrator). This is needed to open the "*Local Group Policy Editor*".
3. Enter "gpedit.msc" to open the "Local Group Policy Editor".

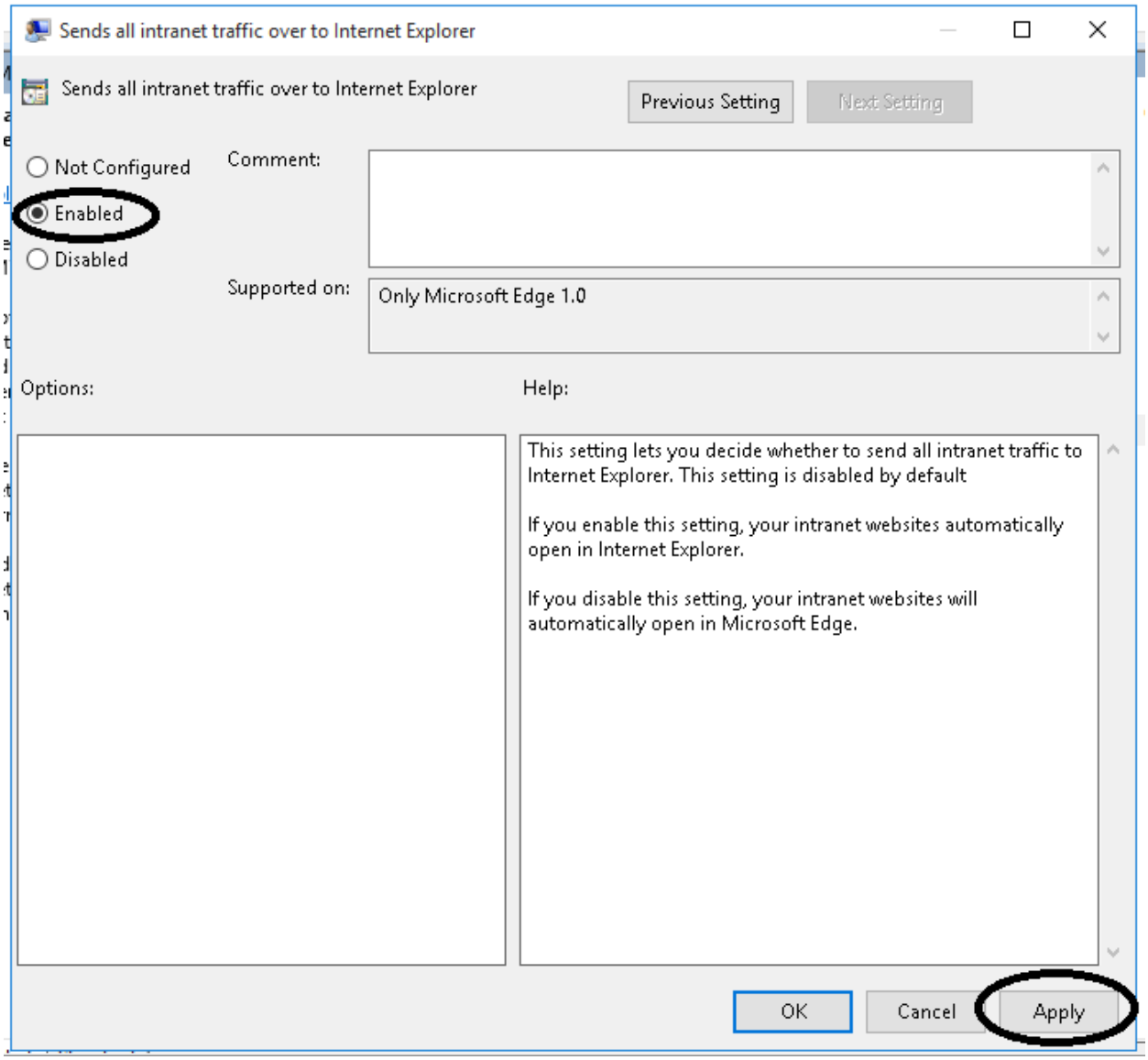


4. Navigate to the Microsoft Edge Folder (Local Computer Policy → Computer Configuration → Administrative Templates → Windows Components → Microsoft Edge).



5. Double Click the Setting "*Sends all intranet traffic over to Internet Explorer*"

6. Enable the feature, click Apply and then OK.



Once this setting is applied, requests to the Insbridge Application will open an IE11 browser. All other intranet traffic will also be sent to IE. This may include and internal company sites.

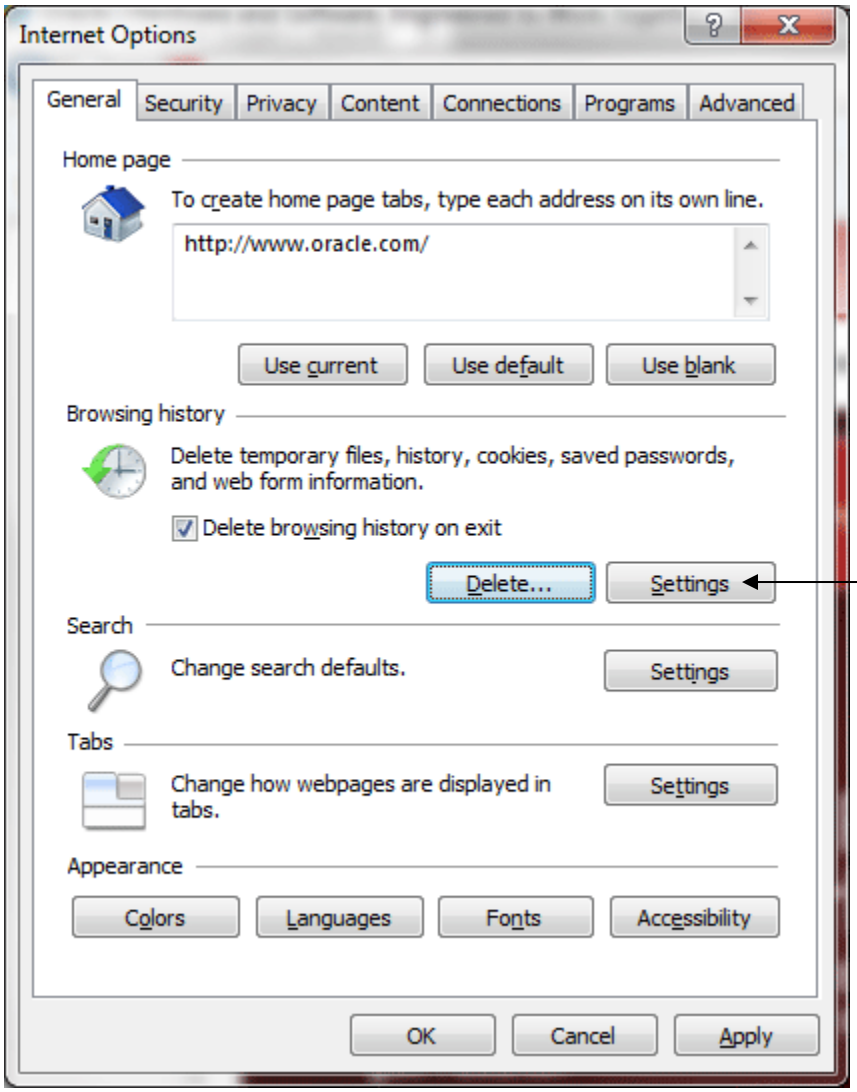
To undo the change, set the Sends all intranet traffic over to Internet Explorer to Not Configured.

Clearing Temp Files

Because Insbridge is a web-based application, you may need to clear out your browser's temporary file periodically. Keeping the temporary file clear may improve performance by clearing off space from your hard drive.

1. Open up a regular browser window, not a RateManager screen. On the top menu bar, click **Tools→Internet Options...**

2. Click Delete… in the Browsing History section of the General Tab, located in the middle section.



To view your temp files or to define your temp file settings, click **SETTINGS**. This opens an information screen where you can make adjustments.

Figure 608 Delete Files

The Delete Browsing History is displayed.

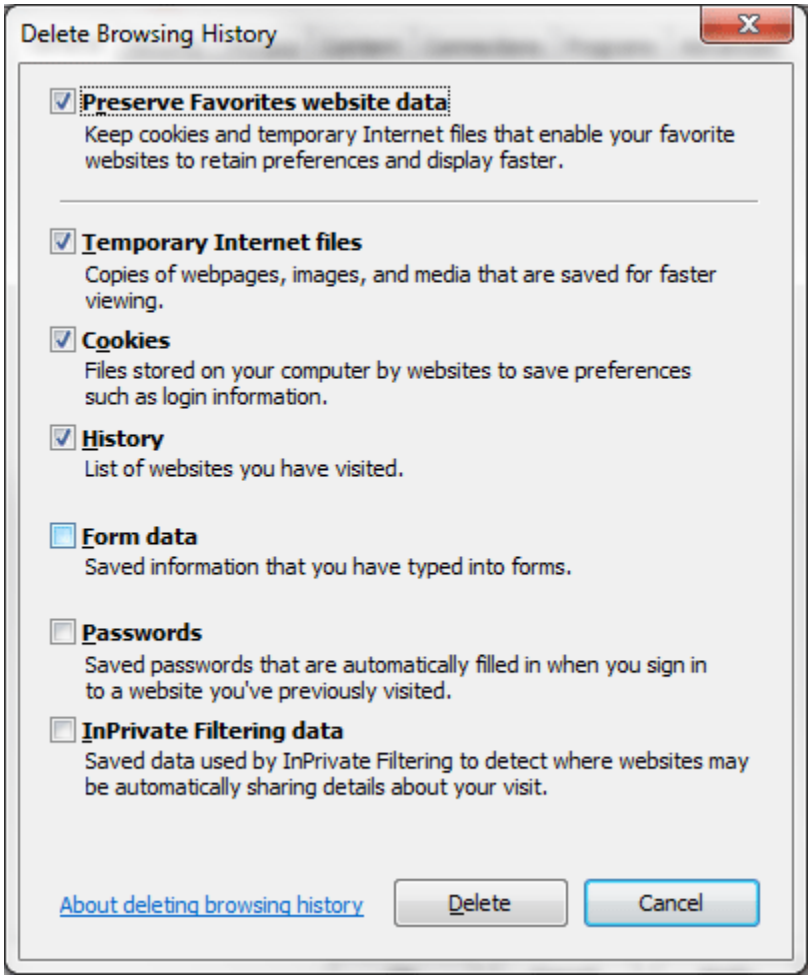


Figure 609 Selecting Files to Delete

3. You can select to delete other content in addition to the temp files.

   Removing stored pages forces the browser to get the page. This may result in slower page loading. However, deleting older pages will help to make sure that you are viewing the most current version of a page.

4. Check the content you want to delete.

5. Click Delete to clear the temp files and other browser history.

   Depending upon the amount of files that need to be deleted, this may take a few minutes. When the temp files have been removed, you will be returned to the previous page.

6. Click **OK** to close the Internet Options box and continue working.

If you do not want to clear your temp file, click Cancel to return to the previous screen.

COUNTRY CODES

NAME	ABBR
AFGHANISTAN	AFG
ALBANIA	ALB
ALGERIA	DZA
AMERICAN SAMOA	ASM
ANDORRA	AND
ANGOLA	AGO
ANGUILLA	AIA
ANTARCTICA	ATA
ANTIGUA AND BARBUDA	ATG
ARGENTINA	ARG
ARMENIA	ARM
ARUBA	ABW
AUSTRALIA	AUS
AUSTRIA	AUT
AZERBAIJAN	AZE
BAHAMAS	BHS
BAHRAIN	BHR
BANGLADESH	BGD
BARBADOS	BRB
BELARUS	BLR
BELGIUM	BEL
BELIZE	BLZ
BENIN	BEN
BERMUDA	BMU
BHUTAN	BTN
BOLIVIA	BOL
BOSNIA AND HERZEGOWINA	BIH
BOTSWANA	BWA
BOUVET ISLAND	BVT
BRAZIL	BRA
BRITISH INDIAN OCEAN TERRITORY	IOT
BRUNEI DARUSSALAM	BRN
BULGARIA	BGR
BURKINA FASO	BFA
BURUNDI	BDI
CAMBODIA	KHM

NAME	ABBR
CAMEROON	CMR
CANADA	CAN
CAPE VERDE	CPV
CAYMAN ISLANDS	CYM
CENTRAL AFRICAN REPUBLIC	CAF
CHAD	TCD
CHILE	CHL
CHINA	CHN
CHRISTMAS ISLAND	CXR
COCOS (KEELING) ISLANDS	CCK
COLOMBIA	COL
COMOROS	COM
CONGO	COG
COOK ISLANDS	COK
COSTA RICA	CRI
COTE D'IVOIRE	CIV
CROATIA (local name: Hrvatska)	HRV
CUBA	CUB
CYPRUS	CYP
CZECH REPUBLIC	CZE
DENMARK	DNK
DJIBOUTI	DJI
DOMINICA	DMA
DOMINICAN REPUBLIC	DOM
EAST TIMOR	TMP
ECUADOR	ECU
EGYPT	EGY
EL SALVADOR	SLV
EQUATORIAL GUINEA	GNQ
ERITREA	ERI
ESTONIA	EST
ETHIOPIA	ETH
FALKLAND ISLANDS (MALVINAS)	FLK
FAROE ISLANDS	FRO
FIJI	FJI
FINLAND	FIN
FRANCE	FRA
FRANCE, METROPOLITAN	FXX
FRENCH GUIANA	GUF
FRENCH POLYNESIA	PYF

NAME	ABBR
FRENCH SOUTHERN TERRITORIES	ATF
GABON	GAB
GAMBIA	GMB
GEORGIA	GEO
GERMANY	DEU
GHANA	GHA
GIBRALTAR	GIB
GREECE	GRC
GREENLAND	GRL
GRENADA	GRD
GUADELOUPE	GLP
GUAM	GUM
GUATEMALA	GTM
GUINEA	GIN
GUINEA-BISSAU	GNB
GUYANA	GUY
HAITI	HTI
HEARD AND MC DONALD ISLANDS	HMD
HONDURAS	HND
HONG KONG	HKG
HUNGARY	HUN
ICELAND	ISL
INDIA	IND
INDONESIA	IDN
IRAN (ISLAMIC REPUBLIC OF)	IRN
IRAQ	IRQ
IRELAND	IRL
ISRAEL	ISR
ITALY	ITA
JAMAICA	JAM
JAPAN	JPN
JORDAN	JOR
KAZAKHSTAN	KAZ
KENYA	KEN
KIRIBATI	KIR
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	PRK
KOREA, REPUBLIC OF	KOR
KUWAIT	KWT
KYRGYZSTAN	KGZ
LAO PEOPLE'S DEMOCRATIC REPUBLIC	LAO

NAME	ABBR
LATVIA	LVA
LEBANON	LBN
LESOTHO	LSO
LIBERIA	LBR
LIBYAN ARAB JAMAHIRIYA	LBY
LIECHTENSTEIN	LIE
LITHUANIA	LTU
LUXEMBOURG	LUX
MACAU	MAC
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	MKD
MADAGASCAR	MDG
MALAWI	MWI
MALAYSIA	MYS
MALDIVES	MDV
MALI	MLI
MALTA	MLT
MARSHALL ISLANDS	MHL
MARTINIQUE	MTQ
MAURITANIA	MRT
MAURITIUS	MUS
MAYOTTE	MYT
MEXICO	MEX
MICRONESIA, FEDERATED STATES OF	FSM
MOLDOVA, REPUBLIC OF	MDA
MONACO	MCO
MONGOLIA	MNG
MONTSERRAT	MSR
MOROCCO	MAR
MOZAMBIQUE	MOZ
MYANMAR	MMR
NAMIBIA	NAM
NAURU	NRU
NEPAL	NPL
NETHERLANDS	NLD
NETHERLANDS ANTILLES	ANT
NEW CALEDONIA	NCL
NEW ZEALAND	NZL
NICARAGUA	NIC
NIGER	NER
NIGERIA	NGA

NAME	ABBR
NIUE	NIU
NORFOLK ISLAND	NFK
NORTHERN MARIANA ISLANDS	MNP
NORWAY	NOR
OMAN	OMN
PAKISTAN	PAK
PALAU	PLW
PANAMA	PAN
PAPUA NEW GUINEA	PNG
PARAGUAY	PRY
PERU	PER
PHILIPPINES	PHL
PITCAIRN	PCN
POLAND	POL
PORTUGAL	PRT
PUERTO RICO	PRI
QATAR	QAT
REUNION	REU
ROMANIA	ROM
RUSSIAN FEDERATION	RUS
RWANDA	RWA
SAINT KITTS AND NEVIS	KNA
SAINT LUCIA	LCA
SAINT VINCENT AND THE GRENADINES	VCT
SAMOA	WSM
SAN MARINO	SMR
SAO TOME AND PRINCIPE	STP
SAUDI ARABIA	SAU
SENEGAL	SEN
SEYCHELLES	SYC
SIERRA LEONE	SLE
SINGAPORE	SGP
SLOVAKIA (Slovak Republic)	SVK
SLOVENIA	SVN
SOLOMON ISLANDS	SLB
SOMALIA	SOM
SOUTH AFRICA	ZAF
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	SGS
SPAIN	ESP
SRI LANKA	LKA

NAME	ABBR
ST. HELENA	SHN
ST. PIERRE AND MIQUELON	SPM
SUDAN	SDN
SURINAME	SUR
SVALBARD AND JAN MAYEN ISLANDS	SJM
SWAZILAND	SWZ
SWEDEN	SWE
SWITZERLAND	CHE
SYRIAN ARAB REPUBLIC	SYR
TAIWAN, PROVINCE OF CHINA	TWN
TAJIKISTAN	TJK
TANZANIA, UNITED REPUBLIC OF	TZA
THAILAND	THA
TOGO	TGO
TOKELAU	TKL
TONGA	TON
TRINIDAD AND TOBAGO	TTO
TUNISIA	TUN
TURKEY	TUR
TURKMENISTAN	TKM
TURKS AND CAICOS ISLANDS	TCA
TUVALU	TUV
UGANDA	UGA
UKRAINE	UKR
UNITED ARAB EMIRATES	ARE
UNITED KINGDOM	GBR
UNITED STATES	USA
UNITED STATES MINOR OUTLYING ISLANDS	UMI
URUGUAY	URY
UZBEKISTAN	UZB
VANUATU	VUT
VATICAN CITY STATE (HOLY SEE)	VAT
VENEZUELA	VEN
VIET NAM	VNM
VIRGIN ISLANDS (BRITISH)	VGB
VIRGIN ISLANDS (U.S.)	VIR
WALLIS AND FUTUNA ISLANDS	WLF
WESTERN SAHARA	ESH
YEMEN	YEM
YUGOSLAVIA	YUG

NAME	ABBR
ZAIRE	ZAR
ZAMBIA	ZMB
ZIMBABWE	ZWE

GLOSSARY TERMS

A	
Administrator:	The person designated by your company who has the authority to create and change groups, usernames, passwords and restrictions.
Algorithm:	A sequence of steps used to perform a calculation.
Assigned Driver:	A driver who has been assigned to a vehicle.
Assigned Vehicle:	A vehicle that has been assigned a driver.
Authoring Environment	The physical machine where RateManager is installed.
C	
Calculated Variable:	Calculated Variables are used when a result cannot best be derived from simple data mapping in a table. For example, if age is not passed as an input, but is a criteria needed in determining other factors, you would use a Calculated Variable to calculate driver age from the inputs of effective date and driver date of birth. Calculated Variables look at every node (driver, vehicle, location, etc.) independently and create a result for each. Once a calculated variable is created, the result can be used in any other variable.
Callouts	A set of a single or multiple programs and/or SoftLibraries that allows users to call needed operations at a specific time from either inside the system or outside the system.
Category:	A user defined group of information that defines fields, variables, algorithms and the overall structure of the program.
Criteria:	A field or variable used in a table variable to determine which value to return. Any field or variable can be used as criteria.
Custom XML:	Custom XML is a feature in the Library module where frequently used custom XML mapping templates can be uploaded and then used by any program that uses custom mappings.
D	
Data Type:	The type of data associated with a parameter. RateManager supports three (decimal, integer and string) data types for variables and fields and a fourth (date) for inputs and result variables only.
Date:	A data type supported for fields and result variables only.
Date Mask:	Specifies how SoftRater interprets dates being passed into an input file.
Decimal:	A data type supported for all types of variables and fields. Examples of decimal values are 3.1415, 18 and 0.995.
Default Value:	Used by a mapped variable if no match is found based on the criteria.
Dependency:	When copying a variable, algorithm or driver assignment, any element that requires another element to be present or defined will be listed. All dependencies must be resolved.
Domain Type:	A domain type is a top level is container that defines the data source. There are three domain types: RateManager, Scenario and Other.

Domains:	Domains are classifications that fall in between domain types and projects that can serve as high level containers where projects can be grouped. At least one domain is required for projects but more can be added if needed.
Driver Assignment:	Driver assignment is an auto insurance specific method of assigning a particular driver to a particular vehicle on a policy, based on certain criteria. Criteria often differ on a carrier basis, and sometimes even on a program level.
Driver Assignment Scenario:	A driver assignment scenario is a list of instructions that define the main sequence of operations to properly define the driver assignment logic for a carrier. A scenario is based upon the same algorithm principle used throughout the RateManager software.
E	
Export:	Exports allow users to export all elements of a program from one database to another database or within the same database. Or to export data in tab-delimited or Microsoft Excel form, from a RateManager table to an outside location.
F	
Flag:	A variable that holds a 1 for true and a 0 for false.
Flag Driver Algorithm:	An algorithm used by a Flag Driver Function to flag drivers based on certain criteria.
Flag Driver Function:	A built-in function used in the main driver assignment that defines the scope of the flagging operation. Most require an association with a predefined Flag Driver Algorithm.
Flag Vehicle Algorithm:	An algorithm used by a Flag Vehicle Function to flag vehicles based on certain criteria.
Flag Vehicle Function:	A built-in function used in the main driver assignment that defines the scope of the flagging operation. Most require an association with a predefined Flag Vehicle Algorithm.
Folders:	A RateManager file management system that functions in much the same way as Microsoft Windows Explorer. This multi-level setup allows for an unlimited number of program folders and subfolders to be placed underneath a product.
G	
Global:	An input or variable that is available to all programs under a specific .
Group:	A set of users that have the same access rights.
I	
IBER:	Insbridge Enterprise Rating System. This is the entire system.
IBFA	Insbridge Framework Administrator. IBFA is an administrative tool used to configure Insbridge applications and setup RateManager database connections. IBFA will be located on a Windows Server machine. IBFA/SR-WIN is an Insbridge Framework Administrator/SoftRater for Windows.
IBSS	Insbridge SoftRater Server. IBSS is the administrative tool for the SoftRater engine. The SoftRater engine is a multi-platform component within Insbridge that executes the rules, rating and underwriting instructions as defined by the user in RateManager. IBSS is usually

	located on a Java machine. IBSS/SR-JAVA is an Insbridge SoftRater Server/SoftRater for Java.
IBRU	The previous acronym used by the Insbridge system. The IBRU acronym may still be used in some areas of the system.
If:	If statements are step types that have conditions where the next action is determined by the meeting of failing of the condition. RateManager has seven types of If statements that are available for use in calculated variables and algorithms.
Impact Analysis:	Impact Analysis is a data analysis tool that allows you to compare the effects of a program change against existing rates.
Import:	Import allows users to bring in programs from an outside location into RateManager. Or to import data in tab-delimited form, into a RateManager table.
Input:	An external value that is passed into the rating system.
Integer:	A data type supported for all types of variables and inputs. Examples of integer values are 3, 1859865 and -47.
Interpolation:	An estimated value derived from two known values.
L	
Library:	The Library module is a tab in RateManager that serves as a central repository within RateManager where export programs, template programs, public XML files, custom XML, and table export files are stored and managed.
Linked Variables:	Two or more mapped variables that have been associated with one another because they use the same criteria.
Lock:	A lock will close all associated Variables, Algorithms, Driver Assignments, Sequencing and Result Mappings in a program version from deletions and edits.
Logical Environment	An environment created for a subscriber in IBFA. It defines package location, engine location and database location in addition to several other supporting data items. This environment is used for rating and/or SRP management. Each database connection will have a logical environment.
M	
Mask:	A feature that allows the customer to determine how data should be interpreted.
N	
Naming Standards:	A uniform naming structure applied to elements and programs either system wide or per product.
Normal Rating Algorithm:	The most common type of algorithm. Examples of what it can be used for are: • Determine premiums • Calculate differences in limits being passed into the system vs. limits being rated by the system • Assign tiers
O	
Operator:	A built-in mathematical function used in calculations and comparisons.

P	
Package:	A small file that holds all the RateManager logic for a specific program and version.
Package Location:	A pointer to a location where SoftRater Packages (SRP's) are stored.
Physical Environment:	A physical environment is generally referred to as a physical machine.
Products:	Products are line of business classifications that fall under projects. Products are the holders for one or more folders. This level also holds global elements that can be used by any program in the project.
Program:	A planned group of procedures executed in a specific order to return a rating. Programs in RateManager typically correspond to rate manuals. Programs can be either created by the users or imported.
Projects:	Projects are classifications that fall in between domains and products. Projects allow for a more precise grouping of products or for a broad grouping of products. Universal elements are at the project level and are unique to the project.
Public XML:	Public XML allows users to upload a single program created in an outside source using standard XML format, in to RateManager.
R	
Rank Driver Algorithm:	An algorithm used by a Rank Driver Function to rate drivers based on certain criteria.
Rank Driver Function:	A built-in function used in the main driver assignment that defines the scope and sorting order of the ranking operation. Most require an association with a predefined Rank Driver Algorithm.
Rank Vehicle Algorithm:	An algorithm used by a Rank Vehicle Function to rate vehicles based on certain criteria.
Rank Vehicle Function:	A built-in function used in the main driver assignment that defines the scope and sorting order of the ranking operation. Most require an association with a predefined Rate Vehicle Algorithm.
Rating Environment	The physical machine(s) where SoftRater is installed. This is typically the same as a SoftRater node.
Reconcile	A comparison feature that compares one program version against another version in the same program and generate a report of the differences.
Releases:	The Releases module is a package management area within RateManager. Releases allows you to package programs in batches and also deploy and load packages directly to IBFA in batch.
Restrictions:	Limitations on viewing and editing pages and fields in the system. Restrictions are assigned and changed by the Administrator.
Result:	A value that is passed out of the system after rating.
Result Mapping:	A defined set of results, fields and variables displayed in the output file.
Revision:	A variable specific type of versioning.
RM	RateManager. RateManager is a component within Insbridge that enables users to manage the product definition and modification process, including rating and underwriting logic.
S	

Sequence:	The order in which algorithms run.
SoftLibrary:	A SoftLibrary is a specially developed program that performs a specific task. SoftLibraries may run their own code or call upon other systems to obtain information outside of RateManager, for example, obtaining a credit score.
SoftRater Node	A SoftRater node is either an IBFA (without RateManager) or IBSS instance on a physical environment.
Source:	The source is the creator of a template and will also be the name of the new project.
SR:	SoftRater. The engine that executes the rating, rules and underwriting instructions defined within RateManager. The rating environment for runtime execution and processing of business content. SoftRater can be further defined by the operating system where it has been loaded.
SRP:	SoftRater Packages. A package that holds all the RateManager logic for a specific program and version.
SR-JAVA:	SoftRater for Java. This is also another name for IBSS.
SR-WIN:	SoftRater for Windows. This is also another name for IBFA.
String:	A data type supported for all types of variables and inputs. Examples of string values are "2.718", "The quick brown fox jumps over the lazy dog." and "001".
T	
Tab-delimited:	A type of text file in which columns are separated by tabs. This is a format for importing tables into RateManager.
Table Job Management:	Table Job Management allows users to export table variables in to a Microsoft Excel workbook. With this feature, users create jobs, select one or more table variables, and then submit the job for processing where the variables are exported in to a Microsoft Excel workbook.
Table Variable:	A variable that uses other variables and fields as criteria in determining the appropriate value.
Template:	Templates are exact copies of existing programs within a project that can be from within your own user group, any other user group within the company or even from an outside company.
Testing Module:	The Testing module is the testing tool within RateManager. It allows you to test your rating program during development, without the need to load packages.
U	
Unassigned Driver:	A driver who has not been assigned to a vehicle.
Unassigned Vehicle:	A vehicle that has not been assigned a driver.
Underwriting Algorithm:	A type of algorithm used to determine if a policy meets the requirements of the company.
Universal:	A field or variable that is available to all programs under a specific project.
V	
Variable:	A name used to represent a value that can change. See Variables in the Contents section for more information.

Version:	One of a sequence of copies of a program, each incorporating new modifications.
VFS	Virtual File Servers. Virtual file server management allows you to set up servers that are in different locations where packages can be downloaded.
W	
Wildcard:	An option available for mapped variables that tells RateManager that one or more rows ignore the value passed in for the criteria.
Working Category:	A classification used to define how elements should run.
Worksheet:	Rating Worksheet / Premium Calculation Worksheet a document generated for each policy to convey basic information used to establish individual rates and premiums.
X	
XML ID:	A number automatically assigned by RateManager to identify fields and categories.

INDEX

#

#
Edit Output Group Mapping, 617
Edit Variable Tab, 272
Versioning Criteria, 149, 165

A

Absolute Value
Get Absolute Value Step, 412

Absolute Value Step, 412

Access Denied to all Products
Projects, 119, 124

Activate
Algorithm, 598
Underwriting Rules, 598

Active Revision
Exporting a Table, 313
List of Values, 252

Add Company Name to SoftRater Package Description
Global Versioning Package, 672, 675, 676

Add Criteria
Versioning Criteria, 149, 164

Add Exceptions
Callout Mapping, 578

Add Input/Output Mappings to SoftRater Package
Global Versioning Package, 674

Adding
Change Control, 747
Condidtions, 383
Constants, 410
Custom Value, 750
Elements, 410
Field Variable from the Edit Variable Tab, 225
Fields, 410
Fields from the Edit Calculated Variable Tab, 452
Fields from the Edit Filter Rules Tab, 557
Fields from the Edit Table Variable Tab, 278
Flagging Functions, 514
Insbridge as a Trusted Site, 752
Miscellaneous Functions, 525
New Step, 452, 556
New Step to a Driver Assignment, 527
New Step to an Algorithm, 475
Notes, 739
Parentheses, 385
Program Version Names, 172
Ranking Function, 518
Result Variable from the Edit Variable Screen, 258
Results, 410
Specific Number of Days, Months or Years, 402
Step, 363
Steps, 410
To Worksheet, 632
Variables, 410
Variables from the Edit Algorithm Screen, 475
Variables from the Edit Driver Assignment Screen, 528
Variables to Output Group Mapping, 617
Versioning Criteria, 148, 151, 163, 166
Where Clause, 382
Where Clause Conditions, 383

Adding Programs
Scenario, 84

Adding Tables
Scenario, 88

Addition Precision, 402

Admininstration
Change Control, 747

Advanced Option
Icon, 184, 186, 187

Advanced Options
Edit Variable Tab, 266, 467
Test Case, 693

Advantages
Driver-Vehicle Pairs, 498

Algorithm
Activate, 598
Adding a Ranking Function, 518
Adding New Step to, 475
Adding to Sequencing, 499
Copy Definition Only, 493
Copying Global, 494
Creating New Algorithm, 473
Creating Results Table, 499
Deleting, 495
Deleting Global, 496
Editing, 491
Error Message for Deleting, 496
Flag Driver, 514, 517
Flag Vehicle, 514, 517
Global, Overview, 461
Icon, 186
Local, Overview, 461
Not Executing, 600
Order, 598
Packaging Errors, 676
Rank Vehicle Algorithms, 520
Ranking Driver Algorithms, 520
Ranking Driver Algorithms Example, 521
Ranking Functions Associated With, 518
Resolving Dependencies on Copying, 460
Result Variables, 254
Reviewing, 32

Set Underwriting to Fail, 362
Setting Order, 598
Algorithm Reports
Parentheses, 381
Algorithm Select
Worksheet, 629
Algorithm/Driver Assignment Listings
Available Sequencing, 602
Algorithms
Change Control, 747
Copying, 492
Creating, 473
Data Queries, 436
Driver Assignment, 497
Menu Bar, 464
Navigating, 463
Results from, 462
Worksheet, 628
All Algorithms
Show Listing For, 600
AND Condition
If Steps, 378
Arithmetic
Comparison Operators, 373
RateManager Performing Calculations, 49
Step Type, 410
Arithmetic Comparison Functions
Usage on String Values, 150, 165, 273
Arithmetic Operators, 411
Binary Forms, 411, 412
Setting, 411
Usage, 411
Arithmetic Step, 442, 447, 452, 462, 470, 475, 503, 508, 527, 547, 552, 556
Arithmetic Steps
Alogrithm, 370
Assign 1st Ranked Veh to 1st Ranked Drv, 521
Assign 1st Ranked Veh to 1st Ranked Drv (Seq), 521
Assign 1st Ranked Veh to 1st Ranked Drv (Usage Set), 522
Assign 1st Ranked Veh to 1st Ranked Drv (witj Exclusion), 522
Assign 1st Ranked Veh to 1st Ranked Drv(Seq) (with Exclusion), 522
Assign 1st Ranked Veh to Flag Drv, 522
Assign All Vehs by Princ Op (Drvs NONEXCLUSIVE), 522
Assignment Functions, 434
Assign Flag Drv to all Unassigned Vehs, 522
Assign Flag Drv to first Unassigned Veh, 522
Assign Unassigned Vehs by Princ Op (Drvs EXCLUSIVE), 523
Assignment Functions, 434
Assign Unassigned Vehs by Princ Op (Drvs NONEXCLUSIVE), 523
Assign Veh Usually Driven (All) by Flag Drv, 523
Assign Veh Usually Driven (All) by Ranked Drv (All) (High-to-Low), 523
Assign Veh Usually Driven (All) by Ranked Drv (All) (Low-to-High), 523
Assign Veh Usually Driven (All) by Ranked Drv (High-to-Low), 523
Assign Veh Usually Driven (All) by Ranked Drv (Low-to-High), 524
Assign Veh Usually Driven by Flag Drv, 524
Assign Veh Usually Driven by Ranked Drv (High-to-Low), 524
Assign Veh Usually Driven by Ranked Drv (Low-to-High), 524
Assign Veh Usually Driven by Unassigned Drvs, 524
Assign Vehs by Drv Usage using 1st Ranked Veh, 524
Assign Vehs by Drv Usage using Last Ranked Veh, 524
Assign Vehs Using DA Override Fields, 525
AssignedDriver, 525
Assigning
Program Version Folders, 175
Assigning Categories
Custom XML File, 653, 666
Assignment
Available Functions, 521
Assignment Functions, 521
Assign 1st Ranked Veh to 1st Ranked Drv, 521
Assign 1st Ranked Veh to 1st Ranked Drv (Seq), 521
Assign 1st Ranked Veh to 1st Ranked Drv (Seq) (with Exclusion), 522
Assign 1st Ranked Veh to 1st Ranked Drv (Usage Set), 522
Assign 1st Ranked Veh to 1st Ranked Drv (with Exclusion), 522
Assign 1st Ranked Veh to Flag Drv, 522
Assign All Vehs by Princ Op (Drvs NONEXCLUSIVE), 522
Assign Flag Drv to all Unassigned Vehs, 522
Assign Flag Drv to first Unassigned Veh, 522
Assign Unassigned Vehs by Princ Op (Drvs EXCLUSIVE), 434, 523
Assign Unassigned Vehs by Princ Op (Drvs NONEXCLUSIVE), 523
Assign Veh Usually Driven (All) by Flag Drv, 523
Assign Veh Usually Driven (All) by Ranked Drv (All) (High-to-Low), 523
Assign Veh Usually Driven (All) by Ranked Drv (All) (Low-to-High), 523
Assign Veh Usually Driven (All) by Ranked Drv (High-to-Low), 523
Assign Veh Usually Driven (All) by Ranked Drv (Low-to-High), 524
Assign Veh Usually Driven by Flag Drv, 524

Assign Veh Usually Driven by Ranked Drv (High-to-Low), 524
Assign Veh Usually Driven by Ranked Drv (Low-to-High), 524
Assign Veh Usually Driven by Unassigned Drvs, 524
Assign Vehs by Drv Usage using 1st Ranked Veh, 524
Assign Vehs by Drv Usage using Last Ranked Veh, 524
Assign Vehs Using DA Override Fields, 525
Driver Usage on Vehicles, 524
Set Principal Operator Variable, 523
Assignment Operation, 521
Asterisk
Ranking Algorithm, 518
Wildcard, 275
At least one of the algorithms in the sequencing Packaging Errors, 676
Attaching Files
Notes, 741
Attachments
Preview, 744
Removing from Notes, 745
Attribute
result_def, 621
Attributes
V, 681, 686
Authoring Environment Definition, 767
Automatic Versioning, 178
Override, 178
Override Example, 178
Setting, 179
Available
Arithmetic Operators, 411
Flagging Functions, 515
Available Assignment Functions, 521
Available Mask Options, 274
Available Ranking Functions, 519
Available Variables, 272
Edit Filter Rules, 553
Edit Output Group Mapping, 616
Edit Variable Tab, 271, 278, 342, 448, 471, 510
Available Variables, Functions, and Constants
Assignment Functions, 521
Driver Result, 521
Miscellaneous Functions, 525
Ranking Functions, 518
Scenario Functions, 513
Vehicle Result, 521

B

Banner
VERSION LOCKED, 734
Binary Forms
Arithmetic Operators, 411, 412
Bitwise AND
Arithmetic Operators, 411
Bitwise AND Operations, 410
Bitwise OR
Arithmetic Operators, 411
Bitwise OR Operations, 410
Breaking
Cascading Ranking Operations, 425
Browse
Category, 270, 446, 452, 469, 474, 506, 527, 551, 556
Browse Category, 270, 446, 452, 469, 474, 507, 551, 556
Browser
Compatibility, xxviii

C

Calculated Variable
Icon, 185
Menu Bar, 441
Saving, 452, 475
Calculated Variables, 439
Navigation, 440, 463
Overview, 439
Calculations
Order of Operations, 49
Unicode, 39
Callout Mapping
Add Exception, 578
Delete Exception, 578
Delete Mappings, 578
Field Exception, 579, 588
Menu Bar, 578, 588
Navigation Bar, 579, 588
Output Exception, 588
Refresh, 578, 588
Save, 588
Callout MappingOutput Exception, 579
Callouts, 569
Deleting Program, 590
Deleting SoftLibraries, 590
Implementing, 592
Locale, 51
Navigating To, 574
Properties, 570
Step Type, 592
Used in Steps, 472, 475, 511, 572, 583
Cascading Ranking Operations, 425, 525
Categories, 213
Delete, 214
Menu Bar, 214
Navigation, 213
New, 214
Notes, 214
Sorting by Column Header, 214
Category
Assignment, 211

Being Used, 219
Browsing for, 270, 446, 452, 469, 474, 506, 527, 551, 556
Deleting on Category Listing Screen, 219
Driver-Vehicle, 213
Edit Output Group Mapping, 617
Editing on Category Listing Screen, 216
Element Listing, 184, 185, 186, 187, 188
Example of Usage, 211
Grayed Out, 213
Locked, 213
Mapping Prior to Outputs, 666
Naming, 215
Overview, 211
Parent, 211
Policy, 212, 213
Usage, 211

Category Filter
Edit Output Group Mapping, 615

Category Instance
Data Queries, 436

Category Item
Step, 391

Category Items, 391

Category Listing Screen
Category Name, 214
Deleting Category, 219
Editing Category, 216
Notes, 214
XML ID, 214
XML Path, 214

Category Name
Category Listing Screen, 214

Cell
Editing in Table, 324

Cells
Changing in Data Table, 324, 347

Change Control, 747
Adding, 747
Admin, 747
Deleting, 749
Justification, 747
Tracking ID, 747

Change Password, 61

Change Password Window, 59, 61

Change Step Order, 387
Edit Driver Assignment Screen, 504
Edit Filter Rules, 548
Edit Variable Screen, 443, 467

Changes
Package, 669

Changing
Active Revision of Global Variable, 595
Cells in Data Table, 324, 347
Comparison Operator, 151, 167
Handling NULL Fields, 266
Handling Wildcarded Fields, 266
Sequence Order, 600
Sequencing Order, 606
Step Order, 453, 491, 528, 557
XML ID, 217

Changing Categories
Filter Rules, 558

Characters
Concatenate Two or More Strings or Characters, 427

Classification
Program Explorer, 64

Clear
Filter Rules, 565

Clear Drv Ranking, 525

Clear Drv Ranking (Usage Set), 525

Clear Ranking Step, 425

Clear Search
Sequence Order, 600

Clear Veh Ranking, 525

Clearing
Clipboard Contents, 367
Domains, 113
Projects, 122
Temporary Files, 758

Clipboard
Editing Steps, 364

Clipboard Contents, 367

Column
Op, 151, 167

Column Sorting, 196
Program Sequencing, 601

Comparing
Program Difference Report, 703, 705

Comparison Operator, 373
Changing, 151, 167
Usage in String Values, 150, 165, 273

Completed
Driver-Assignment, 498

Concatenate
Two or More Strings or Characters, 427

Conditions
Adding, 383
Deleting, 385
Editing, 384
Steps, 377

Constant
Icon, 186

Constants, 536
Adding, 410
Available Variables, Functions, and Constants, 513
Copying, 540
Deleting, 542
Functions, 538
Menu Bar, 538
Navigation, 537
Right Click Menu, 538

Copy

Global Algorithm Right Click Menu, 191
Global Driver Assignment Right Click Menu, 192
Global Filter Rules Right Click Menu, 192
Result Variable, 260
Right Click Menu Output Group Mapping, 611
Worksheet, 636, 637
Worksheet Right Click Menu, 626

Copy & Paste
XML Files, 652, 666

Copy and Paste Steps
Rules, 364

Copy Definition and All Data
Driver Assignment, 530
Variables, 281, 456, 559

Copy Definition Only
Algorithm, 493
Driver Assignments, 530
Variables, 281, 456, 559

Copy Linked Variables, 284

Copy Steps
Editing Steps, 364

Copy Variable
Global Result Listing Right Click Menu, 256
Global Variable Listing Right Click Menu, 189, 190, 193

Copying
Algorithms, 492
Constants, 540
Descriptions in a Program, 160
Driver Assignment, 529
Fields, 228
Filter Rules, 558
Global Algorithm, 494
List of Values, 251
Notes, 738
Program, 159
Table Variables, 285

Copying Text, 44

Count Across Category Item Step, 397

Count Across Category Step
All Available, 398
Example All Available, 398
Example Use Current Path, 397
Use Current Path, 397

Country
Versioning Criteria, 149, 165

Create New Program
Scenario, 94

Create Package, 669
Edit Output Group Mapping, 614
Navigation Bar Debug Test Program, 690
Output Mapping, 610
Program ID Number, 669
Sequencing, 600
Worksheet, 628

Create Versioning Package
Right Click Menu Option, 72

Creating
Algorithm for Result Variables, 499
Data Sources, 340
Data Tables, 32
Date Query, 436
Domain Types, 106
Domains, 110
Filter Rules, 555
Import Files, 287, 295
Linked Variables, 279
New Calculated Variable, 473
New Custom Input Mapping, 649
New Driver Assignment, 526
New Output Mapping, 662
New Program, 153
New Program Version, 159
New Result Group, 611
New Result Variable, 257
New Table Variable, 277, 451, 473
New Test Case, 683
New Variable Revision, 593
New Versions Example, 54
Package, 669
PDF, 720
Program Version Folder, 143
Projects, 118
RateManager Package, 671, 672, 674, 676
Scenario, 84
Sequencing, 604
SoftRater Package, 671, 672, 674, 676
Test Case, 686
Underwriting Algorithms, 473
Versioning Package, 673

Criteria
Data Sources, 340
Edit Variable Tab, 272
Max Number of Criteria, 272
Versioning Criteria, 149, 165

Current Category Instance #, 536

Current System Date and Time, 536

Custom Mapping, 668

Custom Value, 750

Custom XML File
Assigning Categories, 653, 666
Mapping Inputs, 650, 652, 664
Mapping Outputs, 663

Custom XML Files
Testing, 692

Custom XML ID
Fields, 225

Cutting Text, 44

D

D/V Output Separate
Output Mapping, 610, 612

DA Driver Results
Ranking Algorithms, 521

DA Vehicle Results

Ranking Algorithms, 521

Data
Deleting Existing Data Prior to Import, 294, 307
Editing in a Table, 322, 343
Exporting Instead of Editing, 322, 343
Interpreting Data Passed into Program, 274
Using Wildcards, 275

Data Elements
Preparation, 31

Data Preparation, 31

Data Queries, 436

Data Query
Creating, 436
Fields, 437

Data Sources, 338
Creating, 340
Criteria, 340
Deleting, 359
Importing Data, 342

Data Table
Changing Cells, 324, 347

Data Type
Element Listing, 184, 185, 187

Data Type Functions, 400

DataTable
Editing a Program Version, 170

Date Addition Step, 402
Addition Precision, 402

Date Difference (days)
Date Difference Step, 403

Date Difference (months)
Date Difference Step, 403

Date Difference (years)
Date Difference Step, 403

Date Difference Step, 402
Date Difference (Days), 403
Date Difference (Months), 403
Date Difference (Years), 403

Date Formats, 51
Locale, 51

Date Function Step, 401

Date Last Modified
Element Listing, 184, 185, 186, 187, 188
Filter Rules, 567

Date Mask
Versioning Criteria, 149, 165

Dates
Being Passed into an Input File, 51

Days
Adding Specific Number of, 402

Debug Report
Hyperlinks, 692
Select Program, 691
Signed and Unsigned Values, 698
Single Report Format, 688
Summary Report Format, 688
Test Program, 680, 688

Debug Report XML
Test Case, 695

Debug Test Program
Create Package, Navigation Bar, 690
Find Next, Navigation Bar, 690
Find Previous, Navigation Bar, 690
Find, Navigation Bar, 690
Re-Rate, Navigation Bar, 690
Select Program, Navigation Bar, 690
View, Navigation Bar, 690

Decimal Precision, 374, 375

Decimal Symbol, 52
Versioning Criteria, 149, 165

Default
Edit Variable Tab, 270, 278, 342
New Result Group, 612
Output Mapping, 610

Default Values
Fields, 53

Definitions
Program Difference Report, 710

Delete
All Rows in Table, 328, 352
Categories, 214
Edit Variable Tab, 266, 273, 274
Folders, 133
Global Driver Assignment Right Click Menu, 192
Global filter rule Right Click Menu, 192
Global Result Listing Right Click Menu, 256
Global Variable Listing Right Click Menu, 189, 190, 191, 193
Right Click Menu Output Group Mapping, 611
Row in Table, 327, 352
Versioning Criteria, 149, 164
Worksheet, 637
Worksheet Right Click Menu, 626

Delete Exceptions
Callout Mapping, 578

Delete Mappings
Callout Mapping, 578

Delete Revision, 596

Delete Row
Program Version, 169

Delete Step
Edit Driver Assignment Screen, 504
Edit Filter Rules, 548
Edit Variable Screen, 443, 467

Delete Term
Edit Driver Assignment Screen, 504
Edit Filter Rules, 548
Edit Variable Screen, 443, 467

Deleting
Algorithm, 495
Algorithm Error Message, 496
Categories in Use, 219
Category on Category Listing Screen, 219
Change Control, 749

Condidtions, 385
Constants, 542
Data Sources, 359
Domain Types, 108
Domains, 115
Driver Assignment, 532
Driver Assignment Error Message, 533
Existing Data Prior to Import, 294, 307
Fields, 231
Filter Rules, 563
Filter Rules In Use, 563, 568
Folder on Manage Folders, 137
Folders, 137
Global Algorithm, 496
Global Driver Assignment, 533
List of Values, 253
Multiple If Statement, 375
Notes, 746
Output Group Options, 622
Parentheses, 386
Program, 156
Program Version, 160
Program Version Folders, 145
Program Versioning Data, 170
Projects, 125
Result Result Variable, 261
Scenario, 99
Step, 388, 390, 453, 491, 528, 557
Variable, 286, 459
Variable Revision, 596
Variables In Use, 459
Versioning Criteria, 151, 167
Where Clause, 383

Denying
Manage Project Relationships, 206

Dependency All Algorithms Above Report, 703

Dependency Report, 703, 716, 718

Deployment Notes
Global Versioning Package, 672, 674, 676

Deprecate
Fields, 229

Description
Edit Output Group Mapping, 617
Edit Variable Tab, 268

Determining
Relevant Data Passed into Program, 275

Digit Grouping Symbol, 52
Versioning Criteria, 149, 165

Disabling
Worksheet, 635

Disadvantages
Driver-Vehicle Pairs, 498

Divide By
Arithmetic Operators, 411

Domain Types
Creating, 106
Deleting, 108
Editing, 107
Other, 104
Overview, 104
Permissions, 104
Right Click Menu, 109
RteManager, 104

Domains, 110
Clearing, 113
Creating, 110
Deleting, 115
Editing, 111
Right Click Menu, 116

DONE
Next Step, 369

Driver
And Vehicle Elements, Separating, 498
Field, 525

Driver a Vehicle is Assigned to
Identifying, 499

Driver Assignment
Adding Flagging Functions, 514
Adding New Step to, 527
Assignment Functions, 521
Copy Definition and All Data, 530
Copy Definition Only, 530
Copying, 529
Creating, 526
Deleting, 532
Deleting Global, 533
Editing, 528
Error Message for Deleting, 533
Global, Overview, 497
Icon, 186
Local, Overview, 497
Menu Bar, 501, 504
Miscellaneous Functions, 525
Navigating, 500
Navigation, 500
Overview, 497
Projects, 117
Saving, 527
Scenario Functions, 513
Sequencing for Auto Program, 607

Driver Assignment Scenario
Ranking Functions, 518

Driver Assignment Scenarios
Step Types, 362

Driver Result
Available Variables, Functions, and Constants, 521
Setting, 521

Driver Usage on Vehicles, 434, 507
Assignment Functions, 524

Driver/Vehicle
Category Rules, 216

Driver/Vehicle Category
Mapping Fields and Outputs, 575, 585

Driver-Assignment
Completed, 498

Driver-Assignment Scenario
Adding Algorithm, 499

DriverID, 525
Drivers Assignment
  Resolving Dependencies on Copying, 460
Driver-Vehicle Category, 213
Driver-Vehicle Pairs, 498

E

Edit
  Custom Value, 750
  Global Result Listing Right Click Menu, 256
  Output Group Mapping, 614
  Right Click Menu Output Group Mapping, 611
  Worksheet Right Click Menu, 626
Edit Algorithms Screen
  Adding Variables, 475
Edit Calculated Variables Tab
  Adding Variables, 452
Edit Data
  Edit Variable Tab, 266
Edit Data Window, 324, 325, 346, 350
Edit Domain Constant, 540
Edit Driver Assignment Screen
  Adding Variables, 528
  Change Step Order, 504
  Delete Step, 504
  Delete Term, 504
  New Step, 504
Edit Driver Assignment Tab
  Save, 504
Edit Field Variable, 227
Edit Filter Rule, 557
  Name, 549
Edit Filter Rules
  Adding Variables, 557
  Available Variables, 553
  Change Step Order, 548
  Delete Step, 548
  Delete Term, 548
  Naming Information, 549
  Navigation Bar, 548
  New Step, 548
  Save, 548
  Steps, 552
  Variable Details, 552
  Working Category, 551
Edit Group
  Output Mapping, 663
Edit Mode
  View Report, 724
Edit Output Group Mapping
  #, 617
  Category, 617
  Category Filter, 615
  Create Package, 614
  Description, 617
  Enabled, 617
  Map Custom XML, 614
  Menu Bar, 614
  Program Output XML, 614
  Result ID, 617
  Save, 614
  Selected Results, 616
  Variable Filter, 615
Edit Program Screen
  Versioning Criteria, 149, 165
Edit Result Variable, 259
Edit Step
  Edit Variable Screen, 443, 467, 504, 548
Edit Table Variables Tab
  Adding Field Variables, 278
Edit Variable Screen, 453
  Adding Result Variable, 258
  Change Step Order, 443, 467
  Delete Step, 443, 467
  Delete Term, 443, 467
  Edit Step, 443, 467, 504, 548
  Navigating to, 453
  Navigation Bar, 443, 467
  New Step, 443, 467
  Steps, 447, 470, 509
  Variable Details, 447, 470, 509
Edit Variable Tab, 280
  #, 272
  Adding Field Variable, 225
  Advanced Options, 266, 467
  Available Variables, 271, 278, 342, 448, 471, 510
  Criteria, 272
  Default, 270, 278, 342
  Delete, 266, 273, 274
  Description, 268
  Edit Data, 266
  Export, 266
  Import, 266
  Interpolate, 273, 278, 342
  Linked Variables Listing, 268
  Mask, 273, 278, 342
  Name, 269, 277, 341, 445, 468, 506
  New Linked Var, 266
  Op, 272
  Revision, 268
  Save, 266, 274, 443, 467
  Saving Variables, 321, 343
  Type, 270, 273, 277, 341, 446, 451
  Variable Information, 269, 444, 468, 507
  View Report, 273
  View Where Used, 273
  Wildcard, 273
  Working Category, 270, 446, 469, 506
Editing
  Algorithm, 491
  Category on Category Listing Screen, 216
  Cell in Table, 324
  Condidtions, 384
  Data in a Table, 322, 343
  Domain Types, 107
  Domains, 111

Driver Assignment, 528
Folder on Manage Folders, 139
Folders, 138
Import Files, 287, 295
Multiple If Statement, 408
Notes, 745
Package, 669
Package Options Override, 676
Parentheses, 386
Program, 155
Projects, 119, 120
Result Group Options, 613
Result ID to Output Group Mapping, 618
Result Result Variables, 259
Scenario, 92
Special Characters, 35
Step, 364
Where Clause, 382
Worksheet, 627

Editing a Downloaded File
Exported Tables, 318

Editing Tables
Scenario, 103

Edition Notice, 2

Editor
Test Case, 182

Element Listing
Category, 184, 185, 186, 187, 188
Data Type, 184, 185, 187
Date Last Modified, 184, 185, 186, 187, 188
Element Type, 184, 186, 187
Folder, 184, 185, 186, 187, 188
Name, 184, 185, 186, 187
Note, 184, 185, 186, 187, 188
Revision, 184, 185, 186, 187
Scrolling, 195
XML ID, 184

Element Locks, 731
Messages, 732
Releasing, 731

Element Revisions
Program, 56

Element Type
Element Listing, 184, 186, 187

Elements
Adding, 410
Change Control, 747
Notes, 738
Separating Driver and Vehicle, 498
Universals, 197

ELSE
Next Step, 369

Empty Field
Wildcard, 275

Enable
New Result Group, 612
Output Mapping, 610

Enabled
Edit Output Group Mapping, 617

Enabling
Worksheet, 635

Environment
Global Versioning Package, 674
Package Screen Options, 671, 674

Equals
Comparing Strings, 373

Error Message
Deleting a Driver Assignment, 533
Deleting an Algorithm, 496

Errors
At least one of the algorithms in the sequencing, 676
Correcting Packaging, 677
Packaging, 676
There are no algorithms in the sequence, 676
There are no results mapped, 676

Example, 409
All Available Count Across Category Item Step, 398
All Available Get Category Item Step, 392
Category Usage, 211
Custom XML File, 643, 662
Flag Vehicle Algorithm, 517
Input File, 642
Mask Options, 275
OR Condition in If Steps, 378
Override Automatic Versioning, 178
Packaging Errors, 677
Pleasure Use Factor, 521
Ranking Driver Algorithms, 521
RateManager Performing Calculations, 49
Results from Algorithms, 462
Revisions, 56
Setup in Program Explorer, 64
Sum Across Category (All Available, 429
Sum Across Category (Use Current Path, 428
Use Current Path Count Across Category Item Step, 397
Use Current Path Get Category Item Step, 391
Use Current Path Set Category Item Step, 394
Versions, 54

ExampleAND Condition in If Steps, 378

Excel
Creating Data Tables, 32

Excel Files
Exported Tables, 318

EXIT LOOP
Next Step, 369

Export
Edit Variable Tab, 266

Export Data, 311

Exported Tables
Editing a Downloaded File, 318
Excel Files, 318
Re-importing, 320

Exporting
Table, 189, 193, 309, 310, 311

Exporting a Table
Overwriting a File, 317
Run, 315
Save, 315
Save and Run, 315
Workbook Name, 317
Worksheet Name, 317

F

Failing
Policy Using Set Underwriting to Fail, 432, 434
Fakepath, 752
Field
Icon, 184
Wildcard, 275
Wildcard Asterisks, 275
Field Exception
Callout Mapping, 579
Field Exceptions
Callout Mapping, 588
Field ID 17, 525
Field ID 367, 525
Field ID 400, 525
Field Requirements
Program, 604
Field Variable
Adding from the Edit Variable Tab, 225
Changes to Program, 227
Fields, 220
Adding, 410
Adding from the Edit Calculated Variables Tab, 452
Adding from the Edit Filter Rules Tab, 557
Adding from the Edit Table Variables Tab, 278
Copying, 228
Custom XML ID, 225
Default Values, 53
Deleting, 231
Deprecate, 229
Driver, 525
DrvUsePctOnVehX, 507
Functions, 221, 233
Menu Bar, 221, 233
Navigation, 221
Policy Level, 147, 163
Restore, 230
Right Click Menu, 223
System Generated, 220
Universals, 220
User Created, 220
Validation Icon, 222
Vehicle, 525
VehPrincipallyDriven, 507, 508
VehUsePctOnDrvX, 507
Fields and Outputs
Driver/Vehicle Category Mapping, 575, 585
File Management, 131
Categories, 213
Program Explorer, 53
Files
Import, 287, 295
Importing for Notes, 741
Filter Policy Where
Filter Rules, 564
Filter Rule
Icon, 184, 185, 186, 187
Filter Rules
Changing Categories, 558
Clear, 565
Copying, 558
Creating, 555
Date Last Modified, 567
Deleting, 563
Deleting Filter Rules In Use, 563, 568
Filter Policy Where, 564
Find by Name, 567
Find by Revision, 567
Menu Bar, 545, 548
Name, 567
Naming Conventions, 549
Navigation, 544
Refresh, 567
Resolving Dependencies on Copying, 460
Revision, 567
Saving, 556
Select Filter Rule, 564
Template Generated, 543
Find
Navigation Bar Debug Test Program, 690
Find by Name
Filter Rules, 567
Find by Revision
Filter Rules, 567
Find Next
Navigation Bar Debug Test Program, 690
Find Previous
Navigation Bar Debug Test Program, 690
Fixed Constant, 536
Flag 1st Ranked Drv, 515
Flag 1st Ranked Veh, 515
Flag All Vehs, 515
Flag Assigned Vehs, 515
Flag Driver
Algorithms, 514
Flag Driver Algorithms, 517
Flag Drvs All, 515
Flag Drvs All(Usage Set), 515
Flag Drvs Assigned, 516
Flag Drvs from Flag Drvs, 516
Flag Drvs Unassigned, 516
Flag Last Ranked Drv, 516
Flag last Ranked Veh, 516
Flag Unassigned Vehs, 516

Flag Veh from Flagged Veh, 517
Flag Vehicle
  Algorithms, 514
Flag Vehicle Algorithms, 517
  Example, 517
Flagging Algorithms, 517
Flagging Functions, 514
  Available, 515
  Flag 1st Ranked Drv, 515
  Flag 1st Ranked Veh, 515
  Flag All Vehs, 515
  Flag Assigned Vehs, 515
  Flag Drvs All, 515
  Flag Drvs All (Usage Set, 515
  Flag Drvs Assigned, 516
  Flag Drvs from Flag Drvs, 516
  Flag Drvs Unassigned, 516
  Flag Last Ranked Drv, 516
  Flag Last Ranked Veh, 516
  Flag Unassigned Vehs, 516
  Flag Veh from Flagged Veh, 517
Folder
  Element Listing, 184, 185, 186, 187, 188
Folder Details
  Program Details, 183
  Program Explorer, 180
Folder Structure
  File Management, 131
Folders, 131
  Creating Program Version Folders, 143
  Delete, 133
  Deleting, 137
  Deleting Program Version Folders, 145
  Editing, 138
  Editing on Manage Folders, 139
  Moving a Folder, 141
  New, 133, 135
  Overview, 131
  Program Version, 131
  Refresh, 133
  Renaming, 136
  Renaming Program Version Folders, 144
  Right Click Menu, 138
  Standard, 131
Full Access, 41
Full Dependency Report, 703
  Worksheet, 628
Functions
  Algorithm Associated With Ranking, 518
  Available Assignment, 521
  Available Ranking, 519
  Available Variables, Functions, and Constants, 513
  Constants, 538
  Fields, 221, 233
  Ranking, 518
  Ranking, Driver Assignment Scenario, 518
  Set Principal Operator Variable, 522
Functions - Program Callouts, 592

G

Gamma
  Step Type, 424
Get Absolute Value Step, 412
Get Category Item Step, 391
  All Available, 392
  Example All Available, 392
  Example Use Current Path, 391
  Use Current Path, 391
Get Length, 426
Step, 426
Get Ranked Category Item Step, 396
Get String Length, 426
Global
  Changing Active Revision, 595
  Copying Algorithm, 494
  Copying Constants, 540
  Creating New Variable Revision, 593
  Deleting Algorithm, 496
  Deleting Constants, 542
  Deleting Driver Assignment, 533
  Locked Algorithm, 496
  Locked Driver Assignment, 533
  Mapping Inputs at Global Level, 645
  Navigating to Constants, 537
  Navigating to Driver Assignment, 500
  Navigating to Fields, 263, 440, 463
  Navigating to Filter Rules, 544
  Overview, 197
  Table Variable, 262
Global Algorithms
  Overview, 461
Global Driver Assignment
  Overview, 497
Global Inputs Listing
  Import Mapping, 668
Global Inputs Listing screen
  Mapping Inputs, 645
Global Inputs Listing Screen
  Map Custom XML, 645
Global Level
  Input Mapping Screen, 645
  Mapping Inputs, 644
Global Mapping
  Identifier when Saving, 657
  Save, 657
Global Result Listing
  Right Click Menu, 256
Global Variable
  Changes to Program, 453, 491, 528, 557
Global Variables
  Locale, 50
Global Versioning Package
  Add Company Name to SoftRater Package Description, 672, 675, 676

Add Input/Output Mappings to SoftRater Package, 674
Deployment Notes, 672, 674, 676
Environment, 674
Package Comments, 671
Publish Type, 671, 672, 676
Globals
Values in Products, 127
Globals Tab, 209
Granting Access
Manage Project Relationships, 202
Grayed Out Category, 213
Greater Than
Specifying Range of Numbers, 373
Grid View
Creating New Test Case, 683
Test Program, 681
Group
Creating New Result, 611
Group Name
Output Mapping, 610

H

Hide Preview Window, 722
Hierarchy
Program Explorer, 62
Hold
Mask Options, 274
Mask Step, 409
Home Key
Usage, 45
Home Tab, 57
Hyperlinks
Debug Report, 692
Program Version Report, 713

I

IBER Definition, 768
IBFA
Adding Libraries, 570
IBFA Definition, 768
IBRU Definition, 769
IBSS Definition, 769
Identifier
Default for Mapping, 656, 667
Identifying
Driver a Vehicle is Assigned to, 499
Vehicle a Driver is Assigned to, 499
IdrvUsePctOnVehX
Field, 507
IF (Normal)
If Steps, 403
IF All (All Available)
If Steps, 404
IF All (Use Current Path)
If Steps, 404
IF Any (All Available)
If Steps, 404
IF Any (Use Current Path)
If Steps, 404
IF No (All Available)
If Steps, 404
IF No (Use Current Path)
If Steps, 404
If Statement
Multiple, 404
IF Statement
Next Step, 369
If Statements, 403
AND Condition, 378
Example AND Condition, 378
Example OR Condition, 378
OR Condition, 378
If Steps
Comparison Operators, 373
IF All (All Available), 404
IF No (All Available), 404
IF Steps
IF (Normal), 403
IF All (Use Current Path), 404
IF Any (All Available), 404
IF Any (Use Current Path), 404
IF No (Use Current Path), 404
Impact Analysis
Custom Input Mapping Requirements, 649, 654
Implementing Callouts, 592
Import
Deleting Existing Data Prior to Import, 294, 307
Edit Variable Tab, 266
Import File, 287, 295
Previewing, 291, 300, 306
Import Files
Columns, 290
Imported Mapping
RateManager, 668
Importing
Mapping into RateManager, 668
Program Versions, 158
Table, 189, 193, 194
Importing a Table, 290, 297, 306
Importing Data
Data Sources, 342
Importing Files for Notes, 741
Incomplete Gamma
Step Type, 424
Input
Preparation for Mapping, 643
Input File

Example, 642
Requirements SoftRater, 642
Saving Mapping, 657
Input Mapping
Custom Mapping, 649
Custom XML Files, 651, 653, 665
Example of Custom XML File, 643
Using Impact Analysis, 649, 654
Input Mapping Screen, 645, 646, 668
Input Variable
View Report, 724
Input XML Report, 703
Inputs
Category Mapping, 654
Mapping, 644, 650, 652, 655, 664
Mapping at Global Level, 645
Mapping at Program Version Level, 646
Packaging Errors, 676
Saving Mapping, 657
Testing Mapping, 667
Inputs XML
Right Click Menu, 714
Viewing, 714
Insbridge Document Viewer
Re-naming, 744
Saving, 710, 712, 728
Insbridge.XML, 642
Insert
Mask Options, 274
Mask Step, 409
New Row in Table, 326, 329, 330, 351
Program Versioning Data, 169
Internet Explorer
Settings, 752
Interpolate
Edit Variable Tab, 273, 278, 342
Overview, 275
Versioning Criteria, 150, 166
Interpreting
Data Passed into Program, 274
In-Use
Icon, 184, 185, 186, 187, 214
IsAlpha
Data Type Functions, 400
IsAlpha Step, 400
IsDate
Data Type Functions, 400
IsDate Step, 400
IsNumeric
Data Type Functions, 400
IsNumeric Step, 401
Item
Count in a Step, 397

J

Justification
Change Control, 747

K

Keyboard Shortcuts, 44
Keystroke
Mask Step, 409

L

Last Updated
Output Mapping, 610
Less Than
Specifying Range of Numbers, 373
Linked Variables, 279
Creating, 279
Rules, 279
Linked Variables Listing
Edit Variable Tab, 268
List of Values, 232
Active Revision, 252
Copying, 251
Deleting, 253
List of Vlaues
Changes to Program, 249
Loading
Rate Changes Via Versioning, 55
Local
Table Variables, 262
Local Algorithms
Overview, 461
Local Driver Assignment
Overview, 497
Local Files
Opening Test Program, 699
Locale, 49
Callouts, 51
Date Formats, 51
Global Variables, 50
Packaging, 50
Selecting, 49
Table Variables, 50
Templates, 50
Updating, 51
Versioning Criteria, 149, 165
Lock Icons, 730
Locked
Global Algorithm, 496
Global Driver Assignment, 533
Icon, 184, 185, 186, 187, 214
Locked Category, 213
Locked Programs
Scenario, 86
Locking, 730

Program Version, 734
Properties, 730
Template Programs, 737
Log Gamma
Step Type, 423
Logical Environment Definition, 769
Login, 40
Failure, 40
Help, 41
Look For Algorithm
Worksheet, 628

M

Main Notes Screen, 740
Manage Folders, 132
Creating a Program Version Folder, 143
Deleting a Folder, 137
Deleting a Program Version Folder, 145
Editing a Folder, 139
Menu Bar, 133
Moving a Folder, 141
Naming a Folder, 136, 144
Navigating To, 132
Renaming a Program Version Folder, 144
Right Click Menu, 133
Screen Details, 133
Manage Products
Locale, 49
Manage Project Relationships
Denying, 206
Granting Access, 202
Permissions, 202
Universals, 200
Map Custom Input XML
Program Listing, 646
Map Custom Output XML
Right Click Menu Output Group Mapping, 611
Map Custom XML
Edit Output Group Mapping, 614
Global Inputs Listing Screen, 645
Program Listing, 664
Right Click Menu Option, 72
Mapped Worksheet, 638
Mapping
Category Prior to Inputs, 654
Category Prior to Outputs, 666
Creating New Output, 662
Default Identifier, 656, 667
Imported, 668
Importing into RateManager, 668
Inputs, 655
Inputs at Global Level, 645
Inputs at Program Version Level, 646
Outputs, 666
Outputs at Program Version Level, 662
Outputs with Custom XML Files, 663
Preparation for Input, 643
Preparation for Output, 662
Save Mapping Input, 657
Save Mapping Output, 667
Saved at Global Level, 668
Testing, 667
Mapping Group
Overriding, 621
Mapping Inputs, 650, 652, 664
Custom XML File, 650, 652, 664
Overview, 644
Mapping Outputs, 663
Custom XML File, 663
Mask
Edit Variable Tab, 273, 278, 342
Program Date, 51
Versioning Criteria, 150, 165
Mask Icon, 150, 166, 273, 278, 342
Mask Options, 274
Examples, 275
Hold, 274
Insert, 274
Remove, 274
Replace, 274
Mask Step, 409
Available Options, 409
Example, 409
Hold, 409
Insert, 409
Keystroke, 409
Remove, 409
Replace, 409
Masking
Interpreting Data Passed into Program, 274
Overview, 274
Menu
Right Click for Product, 673
Menu Bar
Algorithms, 464
Calculated Variable, 441
Callout Mapping, 578, 588
Categories, 214
Constants, 538
Driver Assignment, 504
Driver Assignment, 501
Edit Output Group Mapping, 614
Fields, 221, 233
Filter Rules, 545, 548
Manage Folders, 133
Output Mapping, 610
Program Detail, 181
Program Explorer, 69, 443, 467, 647
Program Versioning Data, 169
Table Variable, 266
Table Variables, 264
Testing, 680
Version Criteria, 149, 164
Merge Program
Scenario, 94
Message

Character Limits, 427
Messages
Element Locks, 732
Minus
Arithmetic Operators, 411
Miscellaneous Functions
Clear Drv Ranking, 525
Clear Drv Ranking (Usage Set), 525
Clear Veh Ranking, 525
Driver Assignment, 525
Months
Adding Specific Number of, 402
Moving
Folder on Manage Folders, 141
Folders, 141
Program, 154
Step, 387
Multiple If Statement, 404
Multiple IF Statement
Deleting, 375
Editing, 408
Multiply By
Arithmetic Operators, 411

N

Name
Edit Filter Rules, 549
Edit Variable Tab, 269, 277, 341, 445, 468, 506
Element Listing, 184, 185, 186, 187
Filter Rules, 567
Naming
On Manage Folders, 136, 144
Naming a Category, 215
Naming Conventions, 33
Filter Rules, 549
Variables, 269, 445, 468, 506
Naming Information
Edit Filter Rules Tab, 549
Naming Restrictions, 37
Natural Log
Step Type, 414
Navigating
Algorithms, 463
Driver Assignment, 500
Edit Variable Screen, 453
Program Details, 181
RateManager, xxviii
Result Variables, 255
Table Variables, 263
To Calculated Variables, 440, 463
To Callouts, 574
To Constants, 537
To Driver Assignment, 500
To Fields, 221
To Filter Rules, 544
To Manage Folders, 132
To Test Program, 679
To Validations, 233
Navigation
Category, 213
Output Mapping, 609
Navigation Bar
Callout Mapping, 579, 588
Edit Filter Rules, 548
Edit Variable Screen, 443, 467
Test Program, 681
New
Categories, 214
Element Revision, 593
Folders, 133, 135
Navigation Bar Test Program, 680
Output Mapping, 610
Worksheet, 624
New Domain Constant, 539
New Field Variable, 224
New Linked Var
Edit Variable Tab, 266
New Linked Variable, 279
New Result Group
Default, 612
Enable, 612
Output Mapping, 611
New Result Variable
Creating, 257
New Step
Edit Driver Assignment Screen, 504
Edit Filter Rules, 548
Edit Variable Screen, 443, 467
Next Step, 369
No Access, 41
No Round
Arithmetic Step, 375
Not Equals
Comparing Strings, 373
Note
Element Listing, 184, 185, 186, 187, 188
Notepad
Editing XML Files, 652, 666
Notes, 738
Adding, 739
Attaching Files, 741
Categories, 214
Category Listing Screen, 214
Copying, 738
Deleting, 746
Editing, 745
Global Result Listing Right Click Menu, 257
Output Mapping, 610
Program Versions, 160
Removing Attachments, 745
Right Click Menu Output Group Mapping, 611
Versioning Criteria, 149, 164
Worksheet Right Click Menu, 626
Notes Detail Screen, 740

NULL Fields
Changing Handling, 266
Handled in RateManager, 266
Number Formats, 52
Numbers
Comparison Operators, 373

O

OBI Publisher, 705, 725
Only Driver Assignment Scenarios
Show Listing For, 600
Only Normal Rating Algorithms
Show Listing For, 600
Only Underwriting Algorithms
Show Listing For, 600
Op
Edit Variable Tab, 272
Versioning Criteria, 150, 165
Op Column, 151, 167
Open
Navigation Bar Test Program, 680
Operand
Replacing, 410
Operations
Ranking Cascading Operations, 525
Operator
Usage, 272
Operators
Setting Arithmetic, 411
Options
Mask Step, 409
Package, 670
Right Click Menu Output Group Mapping, 611
Steps, 361
Worksheet, 636
Worksheet Right Click Menu, 626
OR Condition
If Steps, 378
Order
Algorithms, 598
Underwriting Rules, 598
Order of Operations, 49
Other
Domain Types, 104
Output
Preparation for Mapping, 662
Saving Mapping, 667
Testing Mapping, 667
Output Code
Output Mapping, 610
Output Exception
Callout Mapping, 579
Output Exceptions
Callout Mapping, 588
Output Group Mapping
Adding Variables to, 617
Copy on Right Click Menu, 611
Delete on Right Click Menu, 611
Edit, 614
Edit on Right Click Menu, 611
Map Custom Output XML on Right Click Menu, 611
Mapped Worksheet, 638
Notes on Right Click Menu, 611
Options on Right Click Menu, 611
Program Ouptput XML on Right Click Menu, 611
Result ID, 618
Output Group Options
Deleting, 622
Output Mapping
Create Package, 610
Creating New, 662
Creating New Result Group, 611
D/V Output Separate, 610, 612
Default, 610
Enable, 610
Example of Custom XML File, 662
Group Name, 610
Last Updated, 610
Menu Bar, 610
Navigation, 609
New, 610
New Result Group, 611
Notes, 610
Output Code, 610
Refresh, 610
SoftLibraries, 588
Test Program, 610
Time Stamp, 610
Type, 610
Output Mapping Function, 608
Output Mapping Group
Right Click Menu, 610
Output XML
Worksheet, 633
Outputs
Category Mapping, 666
Mapping, 663, 666
Mapping at Program Version Level, 662
Override Automatic Versioning, 178
Overriding
Default Mapping Group, 621
Editing Package Options, 676
Overview
Assignment Functions, 521
Calculated Variables, 439
Callouts, 569
Category, 211
Domain Types, 104
Driver Assignment, 497
Flagging Algorithms, 517
Flagging Functions, 514
Folders, 131
Global Algorithms, 461
Global Driver Assignment, 497
Globals, 197

Interpolate, 275
Local Algorithms, 461
Local Driver Assignment, 497
Masking, 274
Miscellaneous Functions, 525
New Program Version, 159
Output Mapping Function, 608
Package, 669
Program Version Folders, 131
Programs, 146
Ranking Algorithms, 520
RateManager, 30
Rating Algorithms, 461
Result Variables, 254
Scenario Functions, 513
Sequencing, 598
System Constant, 535
Table Variables, 262
Underwriting Algorithms, 461
Universals, 197
Versioning, 54
Wildcard, 275
Worksheet, 623

Overwriting a File
Exporting a Table, 317

P

P2P Abbreviation
Projects, 119

P2P Links Debug Report
Select Program, 691

Package
Correcting Errors, 677
Create New, 669
Creating, 669
Creating a Versioning Package, 673
Creating New, Mapping, 657, 667
Creating RateManager, 671, 672, 674, 676
Creating SoftRater, 671, 672, 674, 676
Creation Rules, 669
Editing, 669
Environment, 671, 674
Options, 670
Overview, 669
RateManager, 669, 671, 672, 674, 676
SoftRater, 669, 671, 672, 674, 676
Summary, 670
Testing Mapping, 667
Type, 669

Package Comments
Global Versioning Package, 671

Package Location Definition, 770

Packaging
Locale, 50

Packaging Errors, 676
At least one of the algorithms in the sequencing, 676
Example, 677
Resolution, 677
There are no algorithms in the sequence, 676
There are no results mapped, 676

Packaging Process, 670

Page Size
Program Details, 196

Parent Category, 211

Parentheses
Adding, 385
Algorithms Reports, 381
Deleting, 386
Editing, 386
Nested, 380
Steps, 379

Password
Changing, 61

Paste Steps
Editing Steps, 364

Pasting Text, 44

PDF
Creating, 720

PDF Report, 703

Percentage
Driver Usage on Vehicles, 507

Performing
Assignment Operation, 521

Permissions, 41
Domain Types, 104
Manage Project Relationships, 202
Program Explorer, 63
Scenario, 77
Scenario Manager, 81
Scenario User, 82

Physical Environment Definition, 770

Pleasure Use Factor
Example, 521

Plus
Arithmetic Operators, 411

Policy
Set Underwriting to Fail, 432, 434

Policy Category, 212, 213

Policy Level Fields
Version Selection Criteria, 147, 163

Popup Blocker, xxviii

Power
Step Type, 412

Preview Window, 722

Previewing
Import File, 291, 300, 306

pricing_manager_policy_id
Required for Custom, 654

Print
Navigation Bar Testing, 690

Process
Scenario, 83

Product

Right Click Menu, 673
Products
Locale, 49
Rules, 127
Program
Changing a Field Variable, 227
Changing a Global Variable, 453, 491, 528, 557
Changing a List of Values, 249
Changing a Result Variable, 259, 266
Copying, 159
Copying Descriptions, 160
Create Package, 669
Creating New, 153
Creating New Version, 159
Data Preparation, 31
Deleting, 156
Deleting Version, 160
Editing, 155
Element Revisions, 56
Field Requirements, 604
Locking New Version, 734
Mapping Inputs at Program Version Level, 646
Masking Data, 274
Moving, 154
New Version Overview, 159
Packaging Errors, 676
RateManager, 146
Template Generated, 147
Testing, 678
Undoing a Deletion, 156
Undoing a Version Deletion, 161
Unlocking Version, 736
User Created, 147
Versioning, 157
Wildcard Function, 275
Program Callouts
Functions, 592
Program Creation, 76
Program Date Mask, 51
Program Detail
Menu Bar, 181
Program Explorer, 68, 73
Program Detail Folders
Program Explorer, 180
Program Details
Exporting a Table, 309
Folder Detail, 183
Navigating, 181
Page Size, 196
Scenario, 102
Search, 182
Sorting by Column Header, 196
Program Difference Report, 703, 705
Definitions, 710
Running, 706
Viewing, 709, 728
Program Element Revisions, 54
Program Explorer
Classification, 64
File Management, 53
Folder Details, 180
Hierarchy, 62
Menu Bar, 69, 443, 467, 647
Overview, 62
Permissions, 63
Program Detail, 68, 73
Program Details Folders, 180
Programs, 68, 71
Right Click Menus, 71
Scenario, 77
Scenario User, 101
Search, 70
Setup Example, 64
Structure, 68
Unlocking Program Version, 736
Program ID
Create Package, 669
Program Inputs XML, 715
Viewing, 715
Program Listing
Creating New, 153
Map Custom Input XML, 646
Map Custom XML, 664
Program Listing Screen
Copying a Program, 159
Creating New Program, 153
Creating New Program Version, 159
Deleting a Program, 156
Deleting a Program Version, 160
Locking Program Version, 734
Moving a Program, 154
Program Inputs XML, 715
Right Click Menu, 158
Program Name
Versioning Criteria, 149, 165
Program Output XML
Edit Output Group Mapping, 614
Right Click Menu Output Group Mapping, 611
Program Properties, 155
Program Sequencing
Column Sorting, 601
Sorting by Column Header, 601
Program Version
Program Versioning Data, 169
Unlocking Version, 736
Program Version Folders, 171, 173
Assigning, 175
Moving, 141
Overview, 131
Structure, 174
Unassigning, 177
Viewing, 176
Program Version Level
Input Mapping Screen, 646
Mapping Inputs, 644
Output Mapping, 662
Program Version Lock Icon, 734

Program Version Locks, 734
Program Version Name
  Program Versioning Data, 169
Program Version Names, 171
  Adding, 172
Program Version Report, 703, 711
  Details, 713
Program Versioning, 54
Program Versioning Data, 168
  Delete Row, 169
  Deleting, 170
  Insert, 169
  Menu Bar, 169
  Program Information, 169
  Program Name, 169
  Program Version, 169
  Program Version Name, 169
  Save, 169
Program Versions, 158
  Change Control, 747
  Notes, 160
Program XML Report, 703
Programs
  Locale, 49
  Notes, 738
  Program Explorer, 68, 71
  Removing, 590
Projects, 117
  Clearing, 122
  Creating, 118
  Deleting, 125
  Driver Assignment, 117
  Editing, 119, 120
  P2P Abbreviation, 119
  Right Click Menus, 125, 130
  Rules, 117
  Selecting Products, 119
Properties
  Callouts, 570
  Change Control, 747
  Element Locks, 731
  Notes, 738
  Program Version Locks, 734
Publish Scenario, 94
Publish Type
  Global Versioning Package, 671, 672, 676
  RateManager, 671, 672, 674, 676
  SoftRater, 671, 672, 674, 676
  Versioning Package, 674

R

Range of Numbers
  Specifying, 373
Rank Across Category (All Available) (Low-to-High), 425
Rank Across Category (High-to-Low), 425
Rank Across Category (Low-to-High), 425
Rank Across Category Step, 426
  Types, 425
Rank All Drvs (High-to-Low)
  Ranking Functions, 519
Rank All Drvs (High-to-Low) (Usage Set Conditional)
  Ranking Functions, 519
Rank All Drvs (High-to-Low) (Usage Set)
  Ranking Functions, 519
Rank All Drvs (Low-to-High)
  Ranking Functions, 519
Rank All Drvs (Low-to-High) (Usage Set Conditional)
  Ranking Functions, 519
Rank All Drvs (Low-to-High) (Usage Set)
  Ranking Functions, 519
Rank All Drvs vs All Vehicles (Seq)
  Ranking Functions, 519
Rank All Vehs (High-to-Low)
  Ranking Functions, 519
Rank All Vehs (Low-to-High)
  Ranking Functions, 519
Rank Assigned Vehs (High-to-Low)
  Ranking Functions, 520
Rank Assigned Vehs (Low-to-High)
  Ranking Functions, 520
Rank Flagged Drvs (High-to-Low)
  Ranking Functions, 520
Rank Flagged Drvs (Low-to-High)
  Ranking Functions, 520
Rank Unassigned Vehs (High-to-Low)
  Ranking Functions, 520
Rank Unassigned Vehs (Low-to-High)
  Ranking Functions, 520
Ranking, 425
  Algorithm, 520
  Cascading Operations, 525
  Clearing a Step, 425
  Functions, Algorithm Associated With, 518
Ranking Algorithm
  Adding a Ranking Function, 518
  Functions Associated With, 518
Ranking Algorithms
  Overview, 520
Ranking Driver Algorithm, 520
Ranking Functions, 518
  Adding, 518
  Available, 519
  Driver Assignment Scenario, 518
  Rank All Drvs (High-to-Low), 519
  Rank All Drvs (High-to-Low) (Usage Set Conditional), 519
  Rank All Drvs (High-to-Low) (Usage Set), 519
  Rank All Drvs (Low-to-High), 519
  Rank All Drvs (Low-to-High) (Usage Set Conditional), 519
  Rank All Drvs (Low-to-High) (Usage Set), 519
  Rank All Drvs vs All Vehicles (Seq), 519

Rank All Vehs (High-to-Low), 519
Rank Assigned Drvs (Low-to-High), 520
Rank Assigned Vehs (High-to-Low), 520
Rank Flagged Drvs (High-to-Low), 520
Rank Flagged Drvs (Low-to-High), 520
Rank Unassigned Drvs (Low-to-High), 520
Rank Unassigned Vehs (High-to-Low), 520
Ranking Algorithm, 520
Ranking Operations
Breaking Cascading, 425
Cascading, 525
Ranking Vehicle Algorithm, 520
Rate
Navigation Bar Test Program, 680
RateManager
Assigning XML ID, 214
Category, 211
Changing Internet Explorer Settings, 752
Deleting Algorithm, 495
Different Security Levels, 752
Domain Types, 104
Flagging, 514
Handling NULL Fields, 266
Handling Wildcarded Fields, 266
Navigating, xxviii
Overview, 30
Program Versions, 157
Programs, 146
Publish Type, 671, 672, 674, 676
Restrictions, 33
Versioning, 54
RateManager Package, 669, 671, 672, 674, 676
Completion, 670
Rating
Test Case, 684
Rating Algorithms, 461
Rating Environment Definition, 770
Read Only, 41
Re-Calculate Vehicle Usage Step, 434
Types, 434
Recommendations
For Rate Loading, 30
Refresh
Callout Mapping, 578, 588
Callouts, 588
Filter Rules, 567
Folders, 133
Output Mapping, 610
Re-importing
Exported Tables, 320
Releasing
Element Locks, 731
Remove
Callouts, 590
Mask Options, 274
Mask Step, 409
Removing Programs
Scenario, 87
Removing Tables
Scenario, 89, 90
Renaming
Folders, 136
Program Version Folder, 144
Replace
Mask Options, 274
Mask Step, 409
Replacing
Operand, 410
Reports
View Dependency, 716, 718
View History, 719
View Where Used, 729
Viewing, 722
Worksheet, 628, 631
Re-Rate
Navigation Bar Debug Test Program, 690
Resolution
Packaging Errors, 677
Resolving Dependencies
Copying, 460
Restore
Fields, 230
Restrictions
RateManager, 33
Result
Navigating to Variables, 255
Result Group
Creating New, 611
Result Group Listing
Output Mapping, 663
Result Group Options
Editing, 613
Result ID
Edit Output Group Mapping, 617
Result Result Variable
Deleting, 261
Result Variable
Adding from the Edit Variable Screen, 258
Changes to Program, 259, 266
Icon, 185
Right Click Menu, 256
View Report, 724
Result Variable Listing
Copying Variable, 260
Result Variables
Creating, 499
Overview, 254
Result XML
Driver-Vehicle Pairs, 498
Group Results, 610, 612
Results
Adding, 410
No Results Mapped, 676

Results from Algorithms, 462
Results of Step, 371
  Step, 369
Reviewing
  Algorithms, 32
Revision
  Edit Variable Tab, 268
  Element Listing, 184, 185, 186, 187
  Filter Rules, 567
Revisions
  Creating New Revision, 593
  Creating New Variable Revision, 593
  Deleting Filter Rule, 563
  Deleting Variable, 286, 359, 459
  Deleting Variable Revision, 596
  Example, 56
  Show/Hide, 595, 596
Right Click
  Constants, 538
  Copy on Global Algorithm, 191
  Copy on Global Driver Assignment, 192
  Copy on Global Variable Listing, 192
  Copy Variable on Global Result Listing, 256
  Copy Variable on Global Variable Listing, 189, 190, 193
  Delete on Global Driver Assignment, 192
  Delete on Global Result Listing, 256
  Delete on Global Variable Listing, 189, 190, 191, 192, 193
  Edit on Global Result Listing, 256
  Fields, 223
  Global Result Listing, 256
  Notes on Global Result Listing, 257
  Set Active Revision on Global Variable Listing, 190, 191, 192, 193, 194
  Show/Hide Revisions on Global Algorithm, 189, 190, 191, 192, 193
  Table Variable Criteria, 273
  View Dependency Report on Global Algorithm, 191
  View Dependency Report on Global Driver Assignment, 192
  View Dependency Report on Global Filter Rules, 193
  View Dependency Report on Global Variable Listing, 190
  View History on Global Result Listing, 256
  View History on Global Variable Listing, 190, 194
  View History Report, 192
  View History Report on Global Algorithm, 191
  View History Report on Global Filter Rules, 193
  View Quick Report (All Data) on Global Variable Listing, 190, 194
  View Quick Report on Global Algorithm, 191, 193
  View Quick Report on Global Driver Assignment, 192
  View Quick Report on Global Variable Listing, 190, 194
  View Where Used on Global Result Listing, 256
  View Where Used on Global Variable Listing, 190, 194
Right Click Menu
  Create Versioning Package, 72
  Delete, 273
  Domain Types, 109
  Domains, 116
  Folders, 138
  Inputs XML, 714
  Manage Folders, 133
  Map Custom XML, 72
  Output Group Mapping, 610
  Program Explorer, 71
  Program Listing Screen, 158
  Result Variable, 256
  Steps, 372
  Versioning Criteria, 150, 166
  Versioning Package, 673
  View Report, 273
  View Where Used, 273
Right Click Menu Options, 75
Right Click Menus
  Projects, 125, 130
Right Clock
  View PDF on Global Variable Listing, 190, 194
Right Menu
  Worksheet, 626
RM Definition, 770
Round
  Arithmetic Step, 375
Round Up
  Arithmetic Step, 375
Rounding
  In Steps, 374
Row
  Deleting All from Table, 328, 352
  Deleting from Table, 327, 352
  New in Table, 326, 329, 330, 351
Row Count, 322, 344
Rules
  Copy and Paste Steps, 364
  Deleting Algorithms, 496
  Deleting Driver Assignment, 533
  Driver/Vehicle Category, 216
  Flagging Functions, 514
  Linked Variables, 279
  Order Underwriting, 598
  Products, 127
  Program Versioning Data, 168
  Program Versions, 158
  Projects, 117
  Sequencing, 602
  Underwriting Algorithm, 461
  Worksheet, 623
Run
  Exporting a Table, 315
Running
  Program Difference Report, 706

S

Save
  Callout Mapping, 588
  Edit Driver Assignment Tab, 504
  Edit Filter Rules, 548
  Edit Output Group Mapping, 614
  Edit Variable Tab, 266, 274, 443, 467
  Exporting a Table, 315
  Global Mapping, 657
  Navigation Bar Test Program, 680
  Program Versioning Data, 169
  Versioning Criteria, 149, 164
  Worksheet, 628
Save and Run
  Exporting a Table, 315
Save Sequence
  Sequencing, 600
Saving
  Calculated Variable, 452, 475
  Driver Assignment, 527
  Filter Rules, 556
  Insbridge Document Viewer, 710, 712, 728
  Test Case, 685
Scenario
  Adding Programs, 84
  Adding Tables, 88
  Create New Program, 94
  Creating, 84
  Deleting, 99
  Editing, 92
  Editing Tablees, 103
  Locked Programs, 86
  Manager Permissions, 81
  Merge Program, 94
  Permissions, 77
  Process, 83
  Program Details, 102
  Program Explorer, 77
  Publish, 94
  Removing Programs, 87
  Removing Tables, 89, 90
  Setup, 80
  Unlock, 98
  User Permissions, 82
  View Where Used, 91
Scenario Functions, 513
  Available Variables, Functions, and Constants, 513
  Driver Assignment, 513
Scenario User, 101
Scenarios
  Driver Assignment, 497
Screen Details
  Manage Folders, 133
Screen Options
  Environment, 671, 674
Scrolling
  Element Listing, 195
Search
  Program Details, 182
  Program Explorer, 70
  Sequencing, 600
  Test Case, 691
Search Options
  Sequencing, 600
Searching
  SoftLibraries, 588
Security Levels
  RateManager, 752
Select Filter Rule
  Filter Rules, 564
Select Program
  Debug Report, 691
  Debug Report P2P Links, 691
  Navigation Bar Debug Test Program, 690
Selected Results
  Edit Output Group Mapping, 616
Selecting
  Date Formats, 51
  Number Formats, 52
  Sequencing, 604, 619
Selecting Products
  Projects, 119
Separating
  Driver and Vehicle Elements, 498
Sequence
  Changing Order, 600
  Clear Search, 600
Sequencing, 598
  Available Agorithm/Driver Assignments Listings, 602
  Changing Order, 606
  Create Package, 600
  Creating, 604
  Multiple Selection, 604, 619
  No Algorithms in Package, 676
  Rules, 602
  Save Sequence, 600
  Search, 600
  Search Options, 600
  Show Listing For, 600
  Test Program, 600
Server Files
  Opening Test Program, 699
Set Active Revision
  Global Variable Listing Right Click Menu, 190, 191, 192, 193, 194
Set Category Item Step, 394
  Example Use Current Path, 394
  Use Current Path, 394
Set Message Step, 427
Set Principal Operator Variable, 522
  Assignment Functions, 523
Set Ranked Category Item Step, 396
Set String/Message Step, 427
Set Underwriting to Fail, 461

Step Types, 362
Set Underwriting to Fail Step, 432, 434
Setting
Algorithm Order, 598
Arithmetic Operators, 411
Driver Result, 521
Override Automatic Versioning, 179
Settings
Internet Explorer, 752
Setup
Scenario, 80
Shortcuts
Keyboard, 44
Show Listing For
All Algorithms, 600
Only Driver Assignment Scenarios, 600
Only Normal Rating Algorithms, 600
Only Underwriting Algorithms, 600
Sequencing, 600
Show/Hide Revisions, 56, 595, 596
Global Algorithm Right Click Menu, 189, 190, 191, 193
Global Driver Assignment Right Click Menu, 192
Global Filter Rules Right Click Menu, 193
Signed and Unsigned Values
Debug Report, 698
Single Report Format
Debug Report, 688
SoftLibrary
Removing, 590
Searching, 588
SoftRater
Input File Requirements, 642
Passing in Information, 220
Publish Type, 671, 672, 674, 676
RateManager Package, 669
SoftRater Package, 669
SoftRater Node Definition, 771
SoftRater Package, 669, 671, 672, 674, 676
Completion, 670
Environment, 671, 674
SoftRater Packages
Environments, 770
Sort
Categories, 214
Program Sequencing, 601
Sorting
Program Details, 196
Special Characters
Editing, 35
Notes, 739, 747
Restrictions, 33
Specifying
Identifiers, 656, 667
Range of Numbers, 373
Vehicle Usage by VehicleID (Fields:299/368), 508
SR Definition, 771
SR-JAVA Definition, 771
SRP Definition, 771
SR-WIN Definition, 771
Standard Programs, 147
State/Province
Versioning Criteria, 149, 165
Step
Adding, 363, 410
Adding New, 452, 556
Adding New to a Driver Assignment, 527
Adding New to an Algorithm, 475
Arithmetic Operators, 411
Change Step Order, 387
Changing Order, 453, 491, 528, 557
Clearing a Ranking, 425
Clipboard, 364
Conditions, 377
Copy Steps, 364
Decimal Precision, 374
Deleting, 388, 390, 453, 491, 528, 557
DONE, 369
Editing, 364
EXIT LOOP, 369
Get Category Item, 391
Get Ranked Category Item, 396
Instances of a Specific Item, 397
Moving, 387
Nested Parentheses, 380
Next Step, 369
Options, 361
Order of Operations, 49
Parentheses, 379
Paste Steps, 364
Rank Across Category, 426
Re-Calculate Vehicle Usage, 434
Replacing Operand, 410
Results of Step, 369
Right Click Menu, 372
Set Category Item, 394
Set Message, 427
Set Ranked Category Item, 396
Set String/Message, 427
Set Underwriting to Fail, 432, 434
Setting Rounding, 374
String Addition, 427
Sum Across Category, 428
Updating Terms, 372
Where Clause, 375
STEP #
Next Step, 369
Step 1
Calculated Variable, 442, 447, 452, 462, 470, 475, 508
Driver Assignment, 503, 527
Filter Rule, 547
Filter Rules, 552, 556
Step Selection
Worksheet, 630
Step Types
Arithmetic, 410

Data Queries, 436
Gamma, 424
Incomplete Gamma, 424
Log Gamma, 423
Natural Log, 414
Power, 412

Step Variables
Next Step, 367
Results of Steps, 367, 371, 372

Steps
Algorithm, 370
Callouts, 592
Edit Filter Rules, 552
Edit Variable Screen, 447, 470, 509
IF THEN Statements, 403
Set Underwriting to Fail, 461
Using Callouts, 472, 475, 511, 572, 583
Worksheet, 628

String
Comparing, 373
Mask Step, 409

String Addition Step, 427

String Length, 426

String Values
Usage in Arithmetic Comparison Functions, 150, 165, 273

Structure
Program Explorer, 68

Stylesheet
Using Non Insbridge, 668

Sub-tabs
Test Program, 680

Sum across Category (All Available), 429

Sum Across Category (Use Current Path), 428

Sum Across Category Step, 428
Example, 428, 429
Types, 428

Summary
Package, 670

Summary Report Format
Debug Report, 688

System Category – rate
Required for Custom, 654

System Constant, 536
Overview, 535

T

Tab-Delimited File
Exporting a Table, 189, 193, 194, 309, 310, 311
Import, 290

Tab-Delimited Workbook
Tables, 32

Table
Creating for Excel, 32
Deleting All Rows, 328, 352
Deleting Row, 327, 352
Editing Cell, 324
Editing Data, 322, 343
Exporting, 189, 193, 309, 310, 311
Go To, 292, 301
Importing, 189, 193, 194
Inserting New Row, 326, 329, 330, 351
Wildcard, 292

Table Difference Report, 704

Table Variable
Icon, 184
Menu Bar, 264, 266
Row Count, 322, 344

Table Variable Criteria
Right Click Menu, 273

Table Variable Tab
Checked for Wildcard, 275

Table Variables
Data Sources, 338
Exporting a Table, 309
Global, 262
Local, 262
Locale, 50
Navigation, 263
Overview, 262
Universal, 262

Tabs
Globals, 209

Template Created
Program Versions, 158

Template Lock Icon, 737

Template Programs, 147

Templates
Filter Rules, 543
Locale, 50
Locking Template Programs, 737
Variables, 262, 439, 462

Temporary Files, 758

Terms
Updating Steps, 372

Test Case
Advanced Options, 693
Debug Report XML, 695
Rating, 684
Saving, 685
Search, 691
Worksheets, 695

Test Case Editor
Enabling Worksheet, 640, 695

Test Program, 182
Creating New Test Case in Grid View, 683
Creating Test Case in XML View, 686
Debug Report, 688
Edit Output Group Mapping, 614
Grid View, 681
Navigation, 679
Navigation Bar, 681
New, Navigation Bar, 680
Open, Navigation Bar, 680

Opening Server and Local Files, 699
Output Mapping, 610
Print Results, 614
Rate, Navigation Bar, 680
Save, Navigation Bar, 680
Selecting View, 686
Sequencing, 600
Sub-tabs, 680
View, Navigation Bar, 680
Worksheet, 628
XML View, 678, 681

Test Program Tab, 678

Test Scenarios
Test Program, 678

Testing
Custom XML Files, 692
Mapping, 667
Mapping Prerequisites, 667
Menu Bar, 680
Print, Navigation Bar, 690
Testing Mapping, 667

Testing Programs, 678

Text
Cutting and Pasting, 44

THEN
Next Step, 369

There are no algorithms in the sequence
Packaging Errors, 676

There are no results mapped
Packaging Errors, 676

Time Stamp
Output Mapping, 610
Variable, 185

Tracking ID
Change Control, 747

Truncate
Arithmetic Step, 375

Trusted Site
Insbridge, 752

Type
Edit Variable Tab, 270, 273, 277, 341, 446, 451
Output Mapping, 610
Versioning Criteria, 150, 165

Types
Flagging Algorithms, 517
Package, 669
Rank Across Category Step, 425
Re-Calculate Vehicle Usage Step, 434
Sum Across Category Step, 428
Vehicle Usage Options, 507

U

Unassigning
Program Version Folders, 177

Underwriting
Activate Rules, 598
Algorithm Rules, 461
Algorithms, 461
Algorithms, Set Underwriting to Fail, 362
Creating Algorithm, 473
Order Rules, 598
Set Underwriting to Fail Step, 434

Unicode, 39

Universal
Table Variables, 262

Universal Level
Mapping Inputs, 644

Universal Tab, 198
Functions, 199

Universals, 198
Deleting Fields, 231
Deleting List of ValuesFields, 253
Elements, 197
Exporting a Table, 309
Fields, 220
Functions, 199
Manage Project Relationships, 200
Navigating to Fields, 221
Navigating to Validations, 233
Overview, 197
Projects, 117
Validations, 232

Unlock Scenario, 98

Unlocking
Program Version, 736

Update
Password, 61

Updating
Locale, 51

Upload Custom Mapping, 668

Usage
A Driver Usage on Vehicles, 524
Arithmetic Operators, 411
Assignment Set Principal Operator Variable, 523
Driver Usage on Vehicles, 524
Driver-Vehicle Pairs, 498
Home Key, 45
Set Principal Operator Variable, 523

Usage Set, 519
Flag Drvs All, 515

User Created
Program Versions, 158

User Generated
Element Locks, 731

UTF-8
Unicode, 39

V

Valid Values, 232

Validating
XML, 686

Validation
Icon, 187

Indicated on Fields, 222
Unicode, 39

Validations
Navigation, 233

Values
Interpolate, 275

Variable Details
Edit Filter Rules, 552
Edit Variable Screen, 447, 470, 509

Variable Filter
Edit Output Group Mapping, 615

Variable Information
Edit Variable Tab, 269, 444, 468, 507

Variable Selection Criteria
Versioning Criteria, 149, 165

Variables
Absolute Value Step, 412
Adding, 410
Adding from the Edit Algorithms Screen, 475
Adding from the Edit Driver Assignment Screen, 528
Alpha-Numeric Value, 400
Available for Edit Output Group Mapping, 616
Available Variables, Functions, and Constants, 513
Calculated, 439
Change Control, 747
Copy Definition and All Data, 281, 456, 559
Copy Definition Only, 281, 456, 559
Copy Linked Variables, 284
Copying, 285
Copying Result, 260
Creating Linked, 279
Creating New Calculated, 473
Creating New Table, 277, 451, 473
Creating New Variable Revision, 593
Creating Results Table, 499
Date Value, 400
Deleting, 286, 459
Deleting Existing Data Prior to Import, 294, 307
Deleting Variable Revision, 596
Deleting Variables In Use, 459
Get Absolute Value Step, 412
Get Specific Instance, 391
Get Specific Instance After Ranking, 396
Linked, 279
Max Number of Criteria, 272
Naming Conventions, 269, 445, 468, 506
Navigating to Result, 255
New Revision, 593
Numeric Value, 401
Resolving Dependencies on Copying, 460
Result, 254
Sequential Order of Criteria, 272
Set Specific Instance, 394
Set Specific Instance After Ranking, 396
Template Generated, 262, 439, 462
Time Stamp, 185

Vehicle
And Driver Elements, Separating, 498
Field, 525

Vehicle a Driver is Assigned to
Identifying, 499

Vehicle Result
Available Variables, Functions, and Constants, 521

Vehicle Usage by Vehicle Order (Field: 299), 434

Vehicle Usage by Vehicle Order (Field:299), 507

Vehicle Usage by VehicleID (Field: 299/368), 434

Vehicle Usage by VehicleID (Fields:299/368), 508

Vehicle Usage on Drivers, 434, 507
Rules, 507

Vehicle Usage Option
Flag Drvs All (Usage Set), 515, 519

Vehicle Usage Options, 507
Types, 507
Vehicle Usage by Vehicle Order (Field:299), 507
Vehicle Usage on Drivers, 507

VehPrincipallyDriven
Field, 507, 508

VehUsePctOnDrvX
VehUsePField, 507

Version
Deleting a Program Version, 160
Example, 54
Numbering, 157

Version Criteria
Menu Bar, 149, 164

VERSION LOCKED Banner, 734

Versioning
Automatic, 178
Creating a Versioning Package, 673
Override, 178
Overview, 54
Program, 54, 157

Versioning Criteria
#, 149, 165
Add Criteria, 149, 164
Adding, 148, 151, 163, 166
Country, 149, 165
Criteria, 149, 165
Date Mask, 149, 165
Decimal Symbol, 149, 165
Delete, 149, 164
Deleting, 151, 167
Digit Grouping Symbol, 149, 165
Interpolate, 150, 166
Locale, 149, 165
Mask, 150, 165
Notes, 149, 164
Op, 150, 165
Program Name, 149, 165
Right Click Menu, 150, 166
Save, 149, 164
Sequential Order of Criteria, 149, 165
State/Province, 149, 165
Type, 150, 165
Variable Selection Criteria, 149, 165
Wildcard, 150, 166

Versioning Package
Creating, 673
Publish Type, 674
VFS Definition, 772
View
Navigation Bar Debug Test Program, 690
Navigation Bar Test Program, 680
View Dependency Report
Global Algorithm Right Click Menu, 191
Global Driver Assignment Right Click Menu, 192
Global Filter Rules Right Click Menu, 193
Global Variable Listing Right Click Menu, 190
View History, 719
Global Result Listing Right Click Menu, 256
Global Variable Listing Right Click Menu, 190, 194
View History Report, 703
Global Algorithm Right Click Menu, 191
Global Filter Rules Right Click Menu, 193
Right Click Menu, 192
View Input Report, 704
View PDF
Global Variable Listing Right Click Menu, 190, 194
View Quick Report
Global Algorithm Right Click Menu, 191
Global Driver Assignment Right Click Menu, 192
Global Filter Rules right Click Menu, 193
Global Variable Listing Right Click Menu, 190, 194
View Quick Report (All Data)
Global Variable Listing Right Click Menu, 190, 194
View Quick Reports, 703, 722
View Report
Edit Variable Tab, 273
View Reports, 724
View Where Used, 729
Edit Variable Tab, 273
Global Result Listing Right Click Menu, 256
Global Variable Listing Right Click Menu, 190, 194
Scenario, 91
Worksheet, 641
View Where Used Report, 704
Viewing
Inputs XML, 714
Program Difference Report, 709, 728
Program Inputs XML, 715
Program Version Folders, 176
Reports, 722
Virtual File Server Definition, 772

W

Where Clause
Adding, 382
Adding Conditions, 383
Deleting, 383
Editing, 382
Steps, 375
Wildcard
Asterisks, 275
Checked Table Variable Tab, 275
Determining Relevant Data Passed into Program, 275
Edit Variable Tab, 273
Empty Field, 275
Errors, 275
Overview, 275
Versioning Criteria, 150, 166
Wildcarded Fields
Changing Handling, 266
Handled in RateManager, 266
Workbook Name
Exporting a Table, 317
Working Category
Available Categories, 215
Edit Filter Rules, 551
Edit Variable Tab, 270, 446, 469, 506
Worksheet
Adding Selections, 632
Algorithm Select, 629
Algorithms, 628
Copy, 636, 637
Copy on Right Click Menu, 626
Create Package, 628
Delete, 637
Delete on Right Click Menu, 626
Disabling, 635
Edit on Right Click Menu, 626
Editing, 627
Enabling, 635
Enabling in Test Case Editor, 640, 695
Full Dependency Report, 628
Look For Algorithm, 628
New, 624
Notes on Right Click Menu, 626
Options, 636
Options on Right Click Menu, 626
Output XML, 633
Overview, 623
P2P, 641
Reports, 628, 631
Right Click Menu, 626
Rules, 623
Save, 628
Step Selection, 630
Steps, 628
Test Program, 628
View Where Used, 641
Worksheet Output XML, 628
Worksheet XML on Right Click Menu, 626
worksheet_def_name, 641
Worksheet Name
Exporting a Table, 317
Worksheet Output XML
Worksheet, 628
Worksheet XML
Worksheet Right Click Menu, 626
worksheet_def_name, 641
Worksheets

Test Case, 695

X

XML
Validating, 686
XML Editor
Editing XML Files, 650, 652, 665, 666
XML Files
Copy & Paste, 652, 666
Copying Text, 652, 666
Editors, 652, 666
Example Custom, 643, 662
Pasting Text, 652
Selecting Text, 652, 666
XML Format
Existing, 642
XML ID
Category Listing Screen, 214
Changing, 217
Element Listing, 184
Fields, 225
XML Path
Category Listing Screen, 214
XML Results
Driver-Vehicle Pairs, 498
XML View
Creating Test Case, 686
Test Program, 678, 681

Y

Years
Adding Specific Number of, 402